Разработки интеллектуальной справочной системы по музыкальным произведениям

ОГЛАВЛЕНИЕ

[Перечень условных обозначений](#_Toc230162815)

[Введение](#_Toc230162816)

[1. Технико-экономическое обоснование проектируемой интеллектуальной системы](#_Toc230162817)

[1.1 Актуальность разработки интеллектуальной справочной системы по музыкальным произведениям](#_Toc230162818)

[1.2 Категории пользователей системы](#_Toc230162819)

[1.2.1 Меломан](#_Toc230162820)

[1.2.2 Начинающий слушатель](#_Toc230162821)

[1.2.3 Владелец музыкального магазина или интернет-магазина](#_Toc230162822)

[1.2.4 Исполнитель](#_Toc230162823)

[1.3 Анализ аналогичных систем](#_Toc230162824)

[1.3.1 All Music Guide](#_Toc230162825)

[1.3.2 MusicBrainz!](#_Toc230162826)

[1.3.3 Музыкальный словарь Гроува](#_Toc230162827)

[1.3.4 Сравнение аналогичных систем с интеллектуальным справочником по музыкальным произведениям](#_Toc230162828)

[2. База знаний проектируемой интеллектуальной системы](#_Toc230162829)

[2.1 Задачно-ориентированная спецификация базы знаний проектируемой интеллектуальной системы](#_Toc230162830)

[2.1.1 Тестовый сборник вопросов](#_Toc230162831)

[2.1.2 Список используемых в базе знаний проектируемой интеллектуальной справочной системы ip-компонентов](#_Toc230162832)

[2.2 Онтология базы знаний проектируемой интеллектуальной системы](#_Toc230162833)

[2.3 Содержательная декомпозиция базы знаний проектируемой интеллектуальной системы](#_Toc230162834)

[2.4 Исходные тексты базы знаний прототипа интеллектуальной системы](#_Toc230162835)

[2.5 Верификация и отладка базы знаний прототипа интеллектуальной системы](#_Toc230162836)

[2.6 Спецификация разработанной базы знаний интеллектуальной системы и сертификация разработанных ip-компоненто](#_Toc230162837)

[3. Машина обработки знаний проектируемой интеллектуальной справочной системы](#_Toc230162838)

[3.1 Задачно-ориентированная спецификация машины обработки знаний проектируемой интеллектуальной справочной системы](#_Toc230162839)

[3.1.1 Тестовый список задач](#_Toc230162840)

[3.1.2 Содержательная классификация задач](#_Toc230162841)

[3.1.3 Список используемых в операциях ip-компонентов](#_Toc230162842)

[3.1.4 Классификация и спецификация операций проектируемой машины обработки знаний](#_Toc230162843)

[3.1.5 Декомпозиция операций на подпрограммы и содержательная структура библиотеки программ специфицированных операций](#_Toc230162844)

[3.2 Алгоритмы и исходные тексты программ, реализующие операции машины обработки знаний прототипа интеллектуальной справочной системы](#_Toc230162845)

[3.3 Верификация и отладка программ специфицированных операций](#_Toc230162846)

[3.4 Спецификация разработанных операций и библиотеки программ, сертификация разработанных ip-компонентов](#_Toc230162847)

[4. Пользовательский интерфейс проектируемой интеллектуальной справочной системы](#_Toc230162848)

[4.1 Задачно-ориентированная спецификация пользовательского интерфейса проектируемой интеллектуальной системы](#_Toc230162849)

[4.1.1 Декомпозиция пользовательского интерфейса](#_Toc230162850)

[4.1.2 Список используемых ip-компонентов пользовательского интерфейса](#_Toc230162851)

[4.2 Проектирование интерфейсных подсистем](#_Toc230162852)

[4.3 Верификация и отладка пользовательского интерфейса](#_Toc230162853)

[4.4 Спецификация разработанного пользовательского интерфейса, сертификация разработанных ip-компонентов](#_Toc230162854)

[5. Интеграция разработанной системы с другими системами](#_Toc230162855)

[6. Направления дальнейшего развития прототипа интеллектуальной справочной системы и пути промышленного изготовления спроектированной интеллектуальной системы](#_Toc230162856)

[7. Вклад в развитие семантических технологий проектирования интеллектуальных систем](#_Toc230162857)

[7.1 Список выявленных ошибок среды проектирования интеллектуальных систем](#_Toc230162858)

[7.2 Список предложений по совершенствованию среды проектирования интеллектуальных систем](#_Toc230162859)

[7.3 Список предложений по совершенствованию семантических технологий проектирования интеллектуальных систем](#_Toc230162860)

[7.4 Создание и использование ip-компонентов](#_Toc230162861)

[Библиографический указатель](#_Toc230162862)

Перечень условных обозначений

МП – музыкальные произведения

БЗ – база знаний

ИСС – интеллектуальная справочная система

МОЗ – машина обработки знаний

ПИ – пользовательский интерфейс

Введение

Независимо от рода деятельности, возраста и пола, любому из нас приходилось сталкиваться с таким понятием как музыка. И если раньше это было мало распространенным, доступным для узкого круга людей, то в наши дни трудно встретить человека, который никогда не слышал «волшебных» звуков музыки. В наши дни можно говорить о целой музыкальной индустрии, людей и средств, в которой задействовано не меньше, чем в автомобилестроении или разработке IT-технологий. Одни зарабатывают на этом деньги: устраивая звукозаписывающие лейблы и студии, предоставляя свои услуги по записи, продюсированию исполнителей; открывают обычные и интернет-магазины, где успешно и не очень занимаются продажей уже готовой продукции (кассеты, компакт-диски, DVD-диски, и даже, для особых «гурманов», виниловые пластинки с композициями различных жанров и направлений). У других музыка – это способ самовыражения: поиск сподвижников, создание группы, а возможно выступление соло, написание текста и сочинение музыки, затем запись на студии, организация концертов. Третьи выступают в роли потребителей: скупают компакт-диски с альбомами любимых групп и исполнителей, посещают концерты, чтобы вживую услышать кумиров. Не стоит забывать и про тех, для кого музыка это просто приятный фон дома, на работе или в авто, который помогает скрасить серые будни мегаполисов.

Хочу обратить внимание на термины: звукозаписывающие лейблы, компакт-диски, виниловые пластинки, группы, направление, жанры и т.д. и т.п.. И если с пониманием что это и для чего оно нужно помогут справиться какая-нибудь музыкальная энциклопедия или вездесущий Google. То, что делать и у кого просить помощи, когда любимая композиция после н-ного раза прослушивания перестает нравиться, а домашняя коллекция прослушана и переслушана вдоль и поперек. Да, песен и исполнителей в мире много. Есть маленькое «но» - их слишком много! Так в электронной версии «Музыкального словаря Гроува» содержится 27000 композиций поп-музыки или более 50 дней непрерывного прослушивания (если допустить, что средняя продолжительность композиции около 3 минут). А ведь жанров и направлений не меньше сотни! Так и целой жизни будет мало, чтобы переслушать все, что сочинил музыкальный гений человека, и выбрать именно то, что подходит по вкусу.

Можно выделить несколько аспектов проблемы выбора и поиска музыкальных произведений:

- большой объем неструктурированной информации, хранящейся в различных местах;

- трудность формализации, а порой и отсутствие, правил описания и оценки музыкальных произведений;

- формулирование запроса для поиска композиции. Как говорил известный герой не менее известного фильма: «А ну, сочини для меня что-нибудь такое, чтоб душа сначала развернулась, а потом обратно завернулась».

Такие тяжеловесы как Microsoft Corp. и Yahoo Inc. тратят миллиарды долларов, пытаясь отобрать у Google хотя бы небольшой сегмент рынка интернет-поиска. Но параллельно этому, растущие нишевые поисковые системы переосмысляют способ поиска информации, разрабатывая инструменты, которые тщательно ищут истинный смысл слов на веб-страницах вместо того, чтобы просто предоставить список сайтов с совпадающими ключевыми словами, эти претенденты надеются создать сервисы, которые помогут пользователям быстрее находить то, что те ищут. Потребители хотят находить информацию быстрее, и они ищут инструменты, которые помогли бы им сделать это. Такие поисковые системы используют семантическую науку, изучающую смысл слов, чтобы производить более релевантный поиск. Дело в том, что уровень контекста позволяет семантическим поисковым системам представлять информацию в соответствии с тем, как обычно думают люди. Семантический поиск представляет интерес, как для частных предпринимателей, так и для крупных компаний. Ключ к успеху семантического поиска в том, чтобы выдать результаты в контексте. Это сложно, так как объём информации всё возрастает. Причина, по которой люди хотят разрабатывать семантические поисковые системы, в том, что это лучший способ нахождения информации. Учитывая, что объём контента растёт всё быстрее, поисковые системы должны быть более эффективны и работать по образцу того, как думают люди. Подобные инструменты поиска показывают связи между различными документами и сайтами, основываясь не на ссылках, а на информации, содержащейся на страницах с помощью технологии под названием «графики семантических кластеров». Она отображает результаты в виде, который напоминает паучью паутину.

Одним из примеров интеллектуальных «помощников» является Семантическая Справочная Система (SRS), на базе которой была построена справочная система по музыкальным произведениям. SRS состоит из трех компонентов:

База знаний (БЗ),

Машина обработки знаний,

Пользовательский интерфейс.

Уточним задачи компонентов:

База знаний необходима для хранения знаний о предметной области в системе.

Машина обработки знаний необходима для осуществления поиска, навигации и обработки знаний.

Пользовательский интерфейс необходим для осуществления взаимодействия пользователя с системой.

Наполнение базы знаний новыми знаниями значительно улучшит прикладную интеллектуальную систему. Пользователь сможет получить более подробную и более полную информацию о данной предметной области.

Разработка машины обработки знаний переведет прикладную систему на качественно новый уровень – уровень интеллектуального решателя задач предметной области.

В результате разработки пользовательского интерфейса появятся новые возможности, связанные с поиском по базе знаний, редактированием базы, что облегчит пользователям работу с прикладной интеллектуальной системой.

Целью работы является разработка проекта интеллектуальной справочной системы.

Задачи курсового проекта:

* технико-экономическое обоснование проектируемой интеллектуальной справочной системы;
* разработка базы знаний проектируемой интеллектуальной справочной системы;
* разработка машины обработки знаний проектируемой интеллектуальной справочной системы;
* разработка пользовательского интерфейса интеллектуальной справочной системы;
* реализация прототипа интеллектуальной справочной системы.

1. Технико-экономическое обоснование проектируемой интеллектуальной системы

1.1 Актуальность разработки интеллектуальной справочной системы по музыкальным произведениям

Людям, которые желают приобрести музыкальный компакт-диск или скачать композицию из Интернета, не всегда удается затратить на это минимум своего свободного времени и средств и добиться при этом максимальной результативности. Информации содержащейся в буклете компакт-диска или полученной от продавца чаще всего недостаточно для осознанного выбора. Данная интеллектуальная справочная система предназначена для того чтобы облегчить процесс выбора для некомпетентного потребителя музыкальной индустрии.

Перечислим наиболее распространенные вопросы, интересующие людей в отношении музыкальных произведений:

Общие вопросы:

Какие музыкальные жанры существуют?

На каких носителях можно приобрести муз.произведения?

Какие типы музыкальных коллективов бывают?

и т.д.

Вопросы для конкретного произведения:

Кто автор текста, музыки?

Кто исполнитель?

В какой альбом входит?

В каком году издано?

и т.д.

Для нескольких произведений, альбомов, исполнителей:

Что общего между этими альбомами?

и т.д.

Поиск музыкальных произведений:

Поиск по описанию текста, музыки.

Поиск по году издания, жанру, рейтингу в чартах.

Поиск по исполнителю, альбому.

Поиск автора и названия по фрагменту композиции.

Разрабатываемая интеллектуальная система ответит на подобные вопросы пользователя. Дополненная база знаний позволит осуществить расширенный и углубленный поиск. Удобный интерфейс позволит интуитивно и просто найти все интересующие пользователя сведения. Именно поэтому разработка справочника является актуальной и востребованной.

1.2 Категории пользователей системы

Разрабатываемая интеллектуальная справочная система имеет обширную базу знаний и множество функциональных возможностей. Вследствие этого она поддерживает работу широкого круга пользователей. Можно выделить следующие категории пользователей:

* меломан
* начинающий слушатель
* владелец музыкального магазина или интернет-магазина
* исполнитель

Далее приведены портреты пользователей системы.

1.2.1 Меломан

Портрет:

В данную категорию пользователей попадают люди, хорошо владеющие предметной областью. Они знают историю музыки, ориентируются в жанрах и направлениях. Могут четко сформулировать поисковый запрос, определенно знают, что ищут. Они имеют возможность добавить новые знания, отредактировать или уточнить уже имеющиеся.

Функциональные возможности:

* добавление и удаление видеоклипов
* добавление и удаление аудиоматериала
* редактирование имеющихся знаний
* просмотр знаний содержащихся в БЗ
* поиск композиций по ключевым словам, по четким критериям

Цели:

* пополнение своих знаний
* поиск музыкальных произведений
* заимствование своего опыта для менее «продвинутых» пользователей

1.2.2 Начинающий слушатель

Портрет:

В данную категорию пользователей входят люди, которые желают найти для себя новую музыку, но возможно схожую с той, что ранее они слушали. Желающие познакомиться ближе с миром музыки, разобраться в ее жанрах и направлениях, открыть для себя новых исполнителей и группы. Начинающий слушатель не может четко сформулировать поисковый запрос, поэтому для него разработаны специальные возможности поиска.

Функциональные возможности:

* просмотр знаний содержащихся в БЗ
* просмотр видеоклипов
* прослушивание музыкальных композиций
* поиск исполнителя по фрагменту музыкального произведения
* поиск композиций по ключевым словам в описании
* поиск композиций по эмоциональной окраске музыки и текста
* поиск композиций по жанрам
* поиск композиций с помощью временной ленты
* получение ответа на нетривиальные вопросы о музыкальных произведениях

Цели:

* получение базовых знаний в предметной области
* помощь в ориентировании по музыкальным произведениям
* поиск музыкальных произведений по различным признакам

1.2.3 Владелец музыкального магазина или интернет-магазина

Портрет:

В данную категорию пользователей входят люди занимающиеся продажей носителей информации с музыкальными произведениями, клипами, записями концертов и выступлений. Для них главное структурировать информацию о продукции, для облегчения дальнейшей работы. Предоставление информации по имеющемуся ассортименту конечным потребителям.

Функциональные возможности:

* просмотр информации об исполнителях, альбомах и композициях
* добавление информации о новых исполнителях, альбомах, композициях
* поиск музыкальных произведений

Цели:

* пополнение базы знаний новой информацией об исполнителях, альбомах и композициях
* предоставлении информации конечному потребителю

1.2.4 Исполнитель

Портрет:

В данную категорию пользователей входят люди, занимающиеся творческим процессом: написанием музыки, текстов – созданием новых музыкальных произведений. Для них будет интересно просмотреть готовые композиции, чтобы ориентироваться в современной музыке.

Функциональные возможности:

* поиск музыкальных произведений
* сравнение музыкальных произведений

Цели:

* ориентирование в современной музыкальной индустрии

1.3 Анализ аналогичных систем

1.3.1 All Music Guide

Крупная онлайновая музыкальная база данных, принадлежащая организации All Media Guide. Данный продукт является энциклопедией с удобным графическим интерфейсом и поиском по ключевым словам, названиям произведений, альбомов, исполнителей, содержит хорошо проработанную БД с большим количеством информации. Язык интерфейса английский.

1.3.2 MusicBrainz!

Проект создания открытой музыкальной энциклопедии. MusicBrainz хранит сведения о музыке, которая когда-либо была записана, а не саму музыку. Все основные данные, хранимые в MusicBrainz (зафиксированные данные об исполнителях, альбомах, треках и т.п.) считаются общественным достоянием, а дополнительные сведения, такие, как данные о модерациях, распространяются под лицензией Creative Commons non-commercial share-and-share-alike. Язык интерфейса английский.

1.3.3 Музыкальный словарь Гроува

Британское справочное издание, посвящённое академической музыке. Под редакцией Лоры Мейси работает постоянно обновляемая и пополняемая сетевая версия словаря. Годовая подписка на пользование ею составляет 300 долларов. Права на издание и онлайн-версию с 2004 г. принадлежат издательству Оксфордского университета, вследствие чего сайт из Grove Music Online был переименован в Oxford Music Online. Включает в себя более 50000 статей и 28000 биографий, внесенных более чем 6000 ученых со всего мира. Язык интерфейса английский.

1.3.4 Сравнение аналогичных систем с интеллектуальным справочником по музыкальным произведениям

В аналогичных системах не предусмотрен диалог пользователя и системы, то есть, нет возможности задать справочникам вопросы. В отличие от них в интеллектуальной справочной системе по музыкальным произведениям такая возможность есть. Возможность диалога с системой отображает ее интеллектуальные способности в полной мере. Она способно ответить на все многообразие вопросов, которые могут возникнуть у пользователя в процессе поиска композиции, альбома или исполнителя.

Так же можно сказать о том, что у интеллектуальной справочной системе довольно простой интерфейс. Пользователь сможет быстро разобраться с методами получения необходимой ему информации.

2. База знаний проектируемой интеллектуальной системы

2.1 Задачно-ориентированная спецификация базы знаний проектируемой интеллектуальной системы

2.1.1 Тестовый сборник вопросов

Описание тестовых вопросов:

Все известные исполнители.

Фотография конкретного исполнителя.

Альбомы выбранного исполнителя.

Изображение обложки альбома.

Сравнение двух альбомов по их характеристикам.

Композиции входящие в выбранный альбом.

Прослушивание композиции.

Просмотр видеоклипа на выбранную композицю.

Поиск альбома по его характеристикам.

Поиск композиции по ее характеристикам.

Описание музыкального жанра.

Виды носителей.

Авторы текста и композиторы.

Перечень демонстрационных вопросов:

Получение всех исполнителей БЗ.

Описание основных характеристик исполнителя.

Просмотр фотографии исполнителя.

Альбомы выбранного исполнителя.

Просмотр обложки альбома.

Описание основных характеристик альбома.

Композиции входящие в данный альбом.

Описание основных характеристик композиции.

Поиск композиций по рейтингу.

2.1.2 Список используемых в базе знаний проектируемой интеллектуальной справочной системы ip-компонентов

В данной интеллектуальной справочной системе использованы исходные разделы база знаний help-системы:

ИсхБз. SC-ядро

ИсхБз. SCg-язык

ИсхБз. SС-язык множеств

ИсхБз. SС-язык отношений

ИсхБз. SС-язык числовых систем

ИсхБз. Логический SС-язык

ИсхБз. Гипермедийный SС-язык

ИсхБз. Дидактический SС-язык

ИсхБз. SС-язык описания ошибок

ИсхБз. Пользовательский интерфейс SRS

Включение этого компонента в проектируемую систему позволяет уменьшить сроки разработки, так как нет необходимости повторного проектирования данных фрагментов.

2.2 Онтология базы знаний проектируемой интеллектуальной системы

Бинарные отношения:

«альбом\*» - бинарное отношение между исполнителем и его альбомами.

«композиция\*» - бинарное отношение между альбомом и композициями в него входящими.

«описание\*» - бинарное отношение между объектом и параметрами, его характеризующими.

«страна\*» - бинарное отношение между исполнителем и страной исполнителя.

«год\*» - бинарное отношение между исполнителем и годом его основания.

«вид\*» - бинарное отношение между исполнителем и его видом (группа, соло и т.д.).

«обложка\*» - бинарное отношение между альбомом и его обложкой.

«год выпуска\*» - бинарное отношение между альбомом и годом его выпуска.

«носитель\*» - бинарное отношение между альбомом и носителем, на котором он был издан.

«тип записи\*» - бинарное отношение между альбомом и типом его записи.

«объем\*» - бинарное отношение между альбомом и его объемом.

«награды\*» - бинарное отношение между альбомом и его наградами.

«музыка\*» - бинарное отношение между композицией и его музыкой.

«текст\*» - бинарное отношение между композицией и его текстом.

«время\*» - бинарное отношение между композицией и его продолжительностью.

«место в чарте\*» - бинарное отношение между композицией и местом в чарте.

«лейбл\*» - бинарное отношение между композицией и лейблом звукозаписи.

«композитор\*» - бинарное отношение между музыкой композиции и его композитором.

«автор\*» - бинарное отношение между текстом композиции и его автором.

«тема\*» - бинарное отношение между текстом композиции и его темой.

Всего: 20 бинарных отношений.

«Исполнители» - группа всех исполнителей.

«Альбомы» - группа всех альбомов.

«Композиции» - группа всех композиций.

«Носители» - группа всех носителей.

«Композиторы»» - группа всех композиторов музыки.

«Авторы» - группа всех авторов текста.

Всего: 6 узлов групп.

2.3 Содержательная декомпозиция базы знаний проектируемой интеллектуальной системы

Базу знаний можно поделить на фрагменты, в которых дается описание исполнителей, альбомов, МП и отношений между ними.

Так при описании исполнителя представлены следующие характеристики: страна\*, год основания\*, вид\*. Пример описания одного из исполнителей представлен на рисунке 2.3.1.

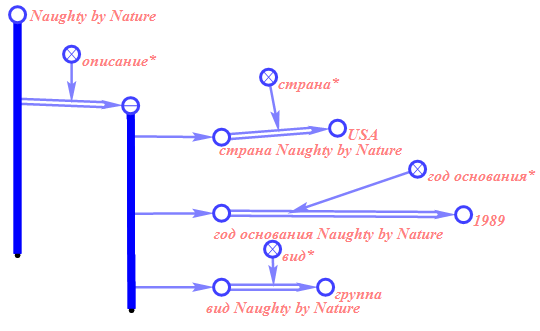


Рисунок 2.3.1. – Описание исполнителя

При описании альбома в качестве его характеристик представлены: обложка альбома\*, год выпуска\*, носитель\*, тип записи\*, объем\*, награды\*. Пример описания альбома представлен на рисунке 2.3.2.

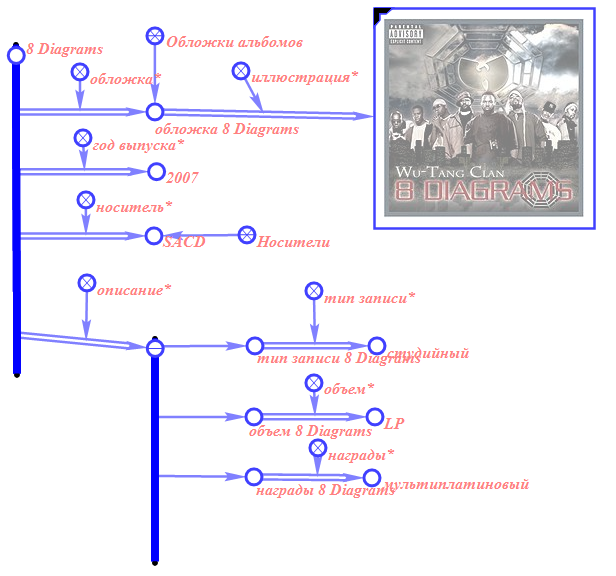


Рисунок 2.3.2. – Описание альбома

Пример описания МП представлен на рисунке 2.3.3.

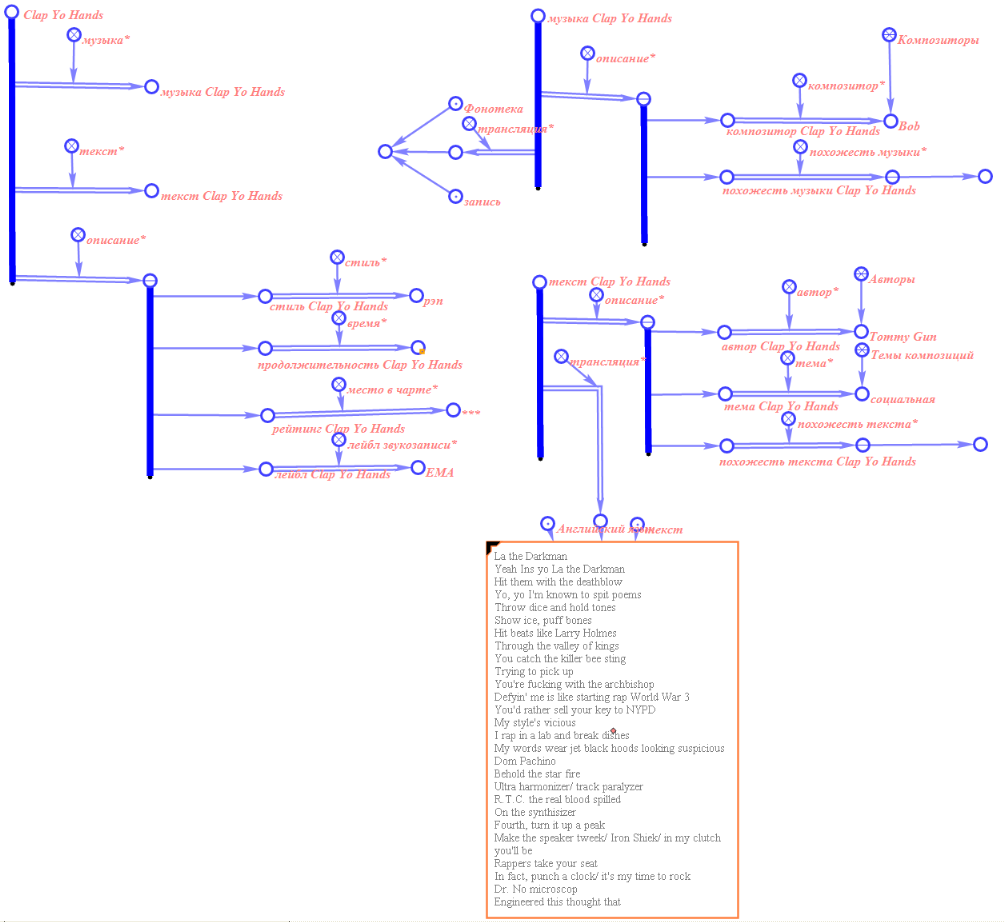


Рисунок 2.3.3. – Описание МП

Пример описания структуры отношений представлен на рисунке 2.3.4.

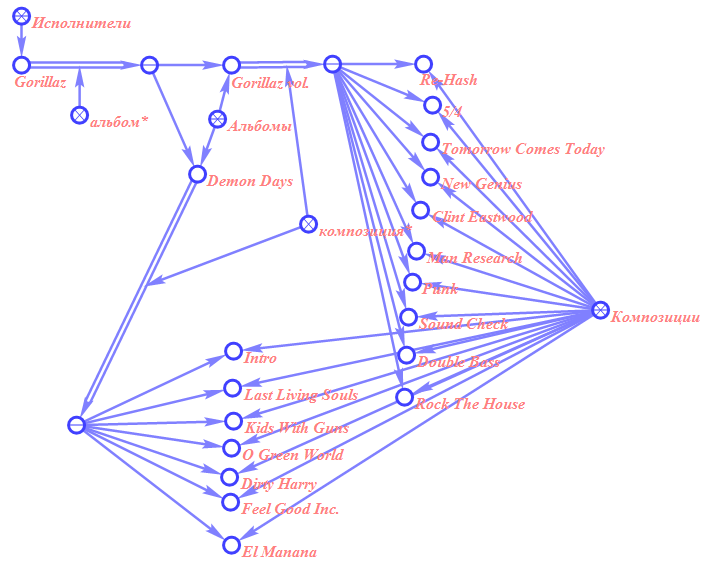


Рисунок 2.3.4. – Описание структуры отношений

2.4 Исходные тексты базы знаний прототипа интеллектуальной системы

Исходные тексты базы знаний прототипа интеллектуальной системы хранятся в следующих файлах:

8diagrams\_album.gwf

bringdaruckus\_track.gwf

clapyohands\_track.gwf

enterthewutang\_album.gwf

familytree\_track.gwf

iicons\_album.gwf

intro.gwf

naughtybynature\_artist.gwf

naughtybynature\_struct.gwf

oldergods\_track.gwf

povetry`sparadise\_album.gwf

protectyaneck\_track.gwf

shameonanigga\_track.gwf

whatyouwannado\_track.gwf

wutangclan\_artist.gwf

wutangclan\_struct.gwf

wutangforever\_album.gwf

elmanana\_track.gwf

feelgoodinc\_track.gwf

ogreenworld\_track.gwf

kidswithguns\_track.gwf

lastlivingsouls\_track.gwf

intro\_track.gwf

demondays\_album.gwf

gorillaz\_struct.gwf

rockthehouse\_track.gwf

doublebass\_track.gwf

soundcheck\_track.gwf

punk\_track.gwf

manresearch\_track.gwf

clinteastwood\_track.gwf

newgenius\_track.gwf

tomorrowcomestoday\_track.gwf

54\_track.gwf

re-hash\_track.gwf

gorillaz\_album.gwf

gorillaz\_artist.gwf

Пример содержания этих файлов можно увидеть на рисунках 2.3.1, 2.3.2, 2.3.3, 2.3.4.

2.5 Верификация и отладка базы знаний прототипа интеллектуальной системы

Верификация базы знаний в текущей версии интеллектуальной справочной системы по МП не производилась, так как в предыдущей версии системы база знаний была грамотно организована. База была дополнена новыми исполнителями, альбомами и композициями. Дополнение и проверка производилась с помощью SRS Studio.

2.6 Спецификация разработанной базы знаний интеллектуальной системы и сертификация разработанных ip-компонентов

В базе знаний на данный момент содержится 3 исполнителя, 7 альбомов и более 20 композиций.

В качестве ip-компонентов могут быть использованы:

Определения понятий (МП, альбом, исполнитель, жанр и т.д.).

Музыкальные композиции.

Видеоклипы к музыкальным произведениям.

3. Машина обработки знаний проектируемой интеллектуальной справочной системы

3.1 Задачно-ориентированная спецификация машины обработки знаний проектируемой интеллектуальной справочной системы

3.1.1 Тестовый список задач

Под предметными задачами будем понимать все задачи, специфичные для данной предметной области:

* поиск исполнителей по описанию
* поиск информации об исполнителях
* поиск альбомов по описанию
* поиск информации об альбомах
* поиск МП по описанию
* сравнение альбомов

3.1.2 Содержательная классификация задач

Множество предметных задач с точки зрения методов их решения можно разбить на два класса:

* информационно-поисковые задачи, ответы на которые присутствуют в базе знаний решателя предметных задач и, следовательно, их надо только найти;
* задачи, ответы на которые требуется построить.

Среди информационно-поисковых задач можно выделить задачи, в которых цель поиска может быть точно описана «шаблоном» поиска, т.е. результатом решения задачи будут все фрагменты семантической сети базы знаний, удовлетворяющие этому шаблону. Примерами таких задач является

* поиск информации о МП;
* поиск информации об альбоме;
* поиск информации об исполнителе.

Другим классом информационно-поисковых задач являются задачи, результат решения которых не описывается шаблоном, а осуществляется с помощью scp-программы. Примерами таких задач являются:

* сравнение альбомов;
* поиск альбомов по нескольким характеристикам;
* поиск МП по нескольким характеристикам.

3.1.3 Список используемых в операциях ip-компонентов

В данной интеллектуально справочной системе были использованы следующие операции, являющиеся Ip-компонентами, и предоставленные вместе со средой разработки интеллектуальных справочных систем:

all\_input

all\_output\_all\_types

analogy

example

graph

lib

logic\_queries

logical\_hierarhy

paramsProcessing

printSetArcs

printSetElems

main\_close

main\_open

open

pack\_menu

peresechenie

reduce\_menu

remove\_menu\_item

searchEnvironmentSmart

searchPatternEx

synonym\_by\_node

synonym\_by\_node\_ui

synonym\_by\_term\_ui

tests

ui\_io\_build\_srs

ui\_io\_copy\_scg

ui\_io\_erase\_scg\_countur

ui\_io\_erase\_scg\_el

ui\_io\_exec\_command

ui\_io\_gen\_srswindow

ui\_io\_increase

ui\_io\_load\_folder

ui\_io\_load\_scg

ui\_io\_modify\_scg\_el\_content

ui\_io\_modify\_scg\_el\_name

ui\_io\_modify\_scg\_el\_type

ui\_io\_paste\_scg

ui\_io\_reduction

ui\_io\_save\_scg

ui\_io\_select\_all

ui\_registration\_type\_wnd

ui\_show\_content

ui\_show\_content\_swf

ui\_unregistration\_type\_wnd

update\_menu

ways

whatisknown

3.1.4 Классификация и спецификация операций проектируемой машины обработки знаний

1. Классификация операции all\_performers.m4scp.

Название операции: all\_performer.

Предметная ориентация: МП.

Входные параметры: нет.

Выходные параметры: все известные исполнители.

Способ вызова: автоматически, при создании окна пользовательского интерфейса.

Спецификация операции all\_performers.m4scp.

Идентификатор операции: descr\_all\_performer.

Полное имя: \operation\ all\_performers.m4scp.

Комментарий: операция поиска в БЗ всех известных исполнителей.

Автор: Лукша Н.В.

2. Классификация операции all\_albums\_this\_performer.m4scp.

Название операции: all\_albums\_this\_performer.

Предметная ориентация: МП.

Входные параметры: узел с именем исполнителя.

Выходные параметры: все альбомы данного исполнителя.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции all\_albums\_this\_performer.m4scp.

Идентификатор операции: descr\_all\_albums\_this\_performer.

Полное имя: \operation\all\_albums\_this\_performer.m4scp.

Комментарий: операция поиска в БЗ всех альбомов данного исполнителя.

Автор: Лукша Н.В.

3. Классификация операции all\_tracks\_this\_album.m4scp.

Название операции: all\_tracks\_this\_album.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: все МП данного альбома.

Способ вызова: при выборе альбома из ListBox.

Спецификация операции all\_tracks\_this\_album.m4scp.

Идентификатор операции: descr\_all\_tracks\_this\_album.

Полное имя: \operation\all\_tracks\_this\_album.m4scp.

Комментарий: операция поиска в БЗ всех МП из данного альбома.

Автор: Лукша Н.В.

4. Классификация операции country\_of\_performer.m4scp.

Название операции: country\_of\_performer.

Предметная ориентация: МП.

Входные параметры: узел с именем исполнителя.

Выходные параметры: страна данного исполнителя.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции country\_of\_performer.m4scp.

Идентификатор операции: descr\_country\_of\_performer.

Полное имя: \operation\country\_of\_performer.m4scp.

Комментарий: операция поиска в БЗ страны данного исполнителя.

Автор: Лукша Н.В.

5. Классификация операции kind\_of\_performer.m4scp.

Название операции: kind\_of\_performer.

Предметная ориентация: МП.

Входные параметры: узел с именем исполнителя.

Выходные параметры: вид данного исполнителя.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции kind\_of\_performer.m4scp.

Идентификатор операции: descr\_kind\_of\_performer.

Полное имя: \operation\kind\_of\_performer.m4scp.

Комментарий: операция поиска в БЗ вида данного исполнителя.

Автор: Лукша Н.В.

6. Классификация операции year\_of\_foundation.m4scp.

Название операции: year\_of\_foundation.

Предметная ориентация: МП.

Входные параметры: узел с именем исполнителя.

Выходные параметры: год основания данного исполнителя.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции year\_of\_foundation.m4scp.

Идентификатор операции: descr\_year\_of\_foundation.

Полное имя: \operation\year\_of\_foundation.m4scp.

Комментарий: операция поиска в БЗ года основания данного исполнителя.

Автор: Лукша Н.В.

7. Классификация операции type\_rec.m4scp.

Название операции: type\_rec.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: тип записи данного альбома.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции type\_rec.m4scp.

Идентификатор операции: descr\_type\_rec.

Полное имя: \operation\type\_rec.m4scp.

Комментарий: операция поиска в БЗ типа записи данного альбома.

Автор: Лукша Н.В.

8. Классификация операции vol.m4scp.

Название операции: vol.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: тип записи данного альбома.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции vol.m4scp.

Идентификатор операции: descr\_vol.

Полное имя: \operation\vol.m4scp.

Комментарий: операция поиска в БЗ типа записи данного альбома.

Автор: Лукша Н.В.

9. Классификация операции style\_of\_track.m4scp.

Название операции: style\_of\_track.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: тип записи данного альбома.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции style\_of\_track.m4scp.

Идентификатор операции: descr\_style\_of\_track.

Полное имя: \operation\style\_of\_track.m4scp.

Комментарий: стиль МП.

Автор: Лукша Н.В.

10. Классификация операции rewards.m4scp.

Название операции: rewards.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: тип записи данного альбома.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции rewards.m4scp.

Идентификатор операции: descr\_rewards.

Полное имя: \operation\rewards.m4scp.

Комментарий: награды данного альбома.

Автор: Лукша Н.В.

11. Классификация операции type\_rec.m4scp.

Название операции: type\_rec.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: тип записи данного альбома.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции type\_rec.m4scp.

Идентификатор операции: descr\_type\_rec.

Полное имя: \operation\type\_rec.m4scp.

Комментарий: операция поиска в БЗ типа записи данного альбома.

Автор: Лукша Н.В.

12. Классификация операции rating.m4scp.

Название операции: rating.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: тип записи данного альбома.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции rating.m4scp.

Идентификатор операции: descr\_rating.

Полное имя: \operation\rating.m4scp.

Комментарий: рейтинг данной композиции.

Автор: Лукша Н.В.

13. Классификация операции label\_rec.m4scp.

Название операции: label\_rec.

Предметная ориентация: МП.

Входные параметры: узел с именем альбома.

Выходные параметры: тип записи данного альбома.

Способ вызова: при выборе исполнителя из ListBox.

Спецификация операции label\_rec.m4scp.

Идентификатор операции: descr\_label\_rec.

Полное имя: \operation\label\_rec.m4scp.

Комментарий: поиск лейбла звукозаписи.

Автор: Лукша Н.В.

3.1.5 Декомпозиция операций на подпрограммы и содержательная структура библиотеки программ специфицированных операций

В написанных scp-операциях подпрограммы не использовались.

3.2 Алгоритмы и исходные тексты программ, реализующие операции машины обработки знаний прототипа интеллектуальной справочной системы

* поиск альбома по описанию (файл search\_by\_param.m4scp)
* входные параметры: узлы, характеризующие искомый альбом;
* выходные параметры: все альбомы подходящие под данное описание.

Алгоритм:

Делаем копию входных параметров.

Проверяем первый параметр.

Находим все альбомы, которые удовлетворяют п.2 и объединяем их в множество 1.

Проверяем второй параметр (если такового нет, переходим п.10).

Находим все альбомы, которые удовлетворяют п.4 и объединяем их в множество 2.

Проверяем третий параметр (если такового нет, переходим в п.9).

Находим все альбомы, которые удовлетворяют п.6 и объединяем их в множество 3.

Находим пересечение множеств 1 и 3 и записываем в множество 1.

Находим пересечение множеств 1 и 2 и записываем в множество 1.

Выводим множество 1 на новый лист.

Пример диалога:

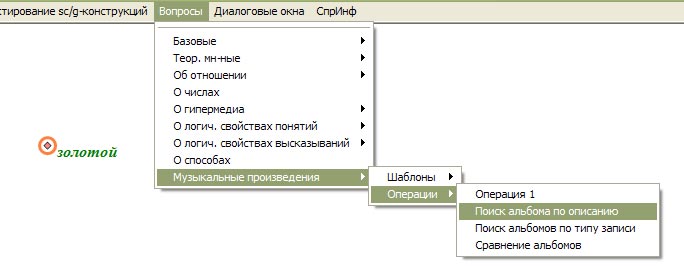


Рисунок 3.2.1 - вопрос «Поиск альбома по описанию»



Рисунок 3.2.2 - ответ на вопрос «Поиск альбома по описанию»

Исходный текст операции:

//#Операция поиска альбома по его описанию

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

procedure(search\_by\_param,

[[

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

sheet\_title = /"Операция прошла успешно"/;

atr1 = "/proc/keynode/1\_";

atr2 = "/proc/keynode/2\_";

ui\_information;

layout;

"MB\_TITLE\_QUERY\_RESULT" = /"Результат запроса"/;

"MB\_TEXT\_METKA" = /"Метка"/;

"MB\_TEXT\_ERROR1" = /"Не равны"/;

"MB\_TEXT\_ERROR2" = /"А сломалося!"/;

"Альбомы";

"описание\*";

"тип записи\*"

]],

[{

sheet,

shift,

node,

node1,

node\_temp,

node\_temp1,

arc,

arc1,

arc2,

attr\_arc,

selected\_els,

selected\_els\_copy,

result\_set,

result\_set1,

result\_set2,

temp\_node\_with\_param,

test\_node,

node\_param,

result,

set\_albums\_copy,

node\_album,

set\_album\_param,

temp\_node,

temp\_set\_1,

temp\_set\_2,

temp\_set\_3,

temp\_set\_5,

temp\_set\_0,

resultSet1,

resultSet2,

resultSet3,

resultSet4,

node\_out\_resSet2,

node\_out\_resSet3,

node\_out\_resSet4,

output\_set

}],

{[

1\_: in\_: selected\_els,

2\_: in\_: sheet,

3\_: in\_: shift

]}

)

genEl([

1\_: assign\_: node\_: output\_set

])

genEl([

1\_: assign\_: node\_: result\_set

])

genEl([

1\_: assign\_: node\_: result\_set1

])

genEl([

1\_: assign\_: node\_: result\_set2

])

//#Копия selected\_els\_copy входных данных

searchSetStr3([

1\_: fixed\_: selected\_els,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: selected\_els\_copy

])

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: temp\_node\_with\_param

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Делаем копию set\_albums\_copy всех альбомов

searchSetStr3([

1\_: fixed\_: "Альбомы",

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: set\_albums\_copy

])

//#Проверяем описание каких альбомов удовлетворяет узел temp\_node\_with\_param

//#Берем один альбом node\_album из множества альбомов set\_albums\_copy

label(other\_album)

searchElStr3([

1\_: fixed\_: set\_albums\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: const\_: node\_album

],,not\_other\_album)

//#Удаляем дугу arc проведенную из set\_albums\_copy к node\_album

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#---------------2\_проходим тройку\_------------------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet3

])

searchSetStr3([

1\_: fixed\_: node\_out\_resSet3,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet4

])

//#Альбомы описываются по трем критериям, проверяем каждый из

//#---------------1\_Проверка по типу записи\_--------------

label(other\_handler)

searchElStr3([

1\_: fixed\_: resultSet4,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet4

],,other\_album)

//#Удаляем дугу arc проведенную из resultSet4 к node\_out\_resSet4

eraseEl([

1\_: fixed\_: arc

])

//#--------Ищем альбом по его введенному описанию---------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_out\_resSet4,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: temp\_node\_with\_param,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set1\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

],,other\_handler)

genElStr3([

1\_: fixed\_: result\_set,

2\_: assign\_: const\_: pos\_: arc,

3\_: fixed\_: node\_album

],other\_album,)

label(not\_other\_album)

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

////

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: temp\_node\_with\_param

],,output)

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Делаем копию set\_albums\_copy всех альбомов

searchSetStr3([

1\_: fixed\_: "Альбомы",

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: set\_albums\_copy

])

//#Проверяем описание каких альбомов удовлетворяет узел temp\_node\_with\_param

//#Берем один альбом node\_album из множества альбомов set\_albums\_copy

label(other\_album1)

searchElStr3([

1\_: fixed\_: set\_albums\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: const\_: node\_album

],,not\_other\_album1)

//#Удаляем дугу arc проведенную из set\_albums\_copy к node\_album

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#---------------2\_проходим тройку\_------------------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet3

])

searchSetStr3([

1\_: fixed\_: node\_out\_resSet3,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet4

])

//#Альбомы описываются по трем критериям, проверяем каждый из

//#---------------1\_Проверка по типу записи\_--------------

label(other\_handler1)

searchElStr3([

1\_: fixed\_: resultSet4,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet4

],,other\_album1)

//#Удаляем дугу arc проведенную из resultSet4 к node\_out\_resSet4

eraseEl([

1\_: fixed\_: arc

])

//#--------Ищем альбом по его введенному описанию---------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_out\_resSet4,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: temp\_node\_with\_param,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set1\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

],,other\_handler1)

genElStr3([

1\_: fixed\_: result\_set1,

2\_: assign\_: const\_: pos\_: arc,

3\_: fixed\_: node\_album

],other\_album1,)

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

label(not\_other\_album1)

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

//

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: temp\_node\_with\_param

],,output)

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Делаем копию set\_albums\_copy всех альбомов

searchSetStr3([

1\_: fixed\_: "Альбомы",

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: set\_albums\_copy

])

//#Проверяем описание каких альбомов удовлетворяет узел temp\_node\_with\_param

//#Берем один альбом node\_album из множества альбомов set\_albums\_copy

label(other\_album2)

searchElStr3([

1\_: fixed\_: set\_albums\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: const\_: node\_album

],,not\_other\_album2)

//#Удаляем дугу arc проведенную из set\_albums\_copy к node\_album

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#---------------2\_проходим тройку\_------------------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet3

])

searchSetStr3([

1\_: fixed\_: node\_out\_resSet3,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet4

])

//#Альбомы описываются по трем критериям, проверяем каждый из

//#---------------1\_Проверка по типу записи\_--------------

label(other\_handler2)

searchElStr3([

1\_: fixed\_: resultSet4,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet4

],,other\_album2)

//#Удаляем дугу arc проведенную из resultSet4 к node\_out\_resSet4

eraseEl([

1\_: fixed\_: arc

])

//#--------Ищем альбом по его введенному описанию---------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_out\_resSet4,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: temp\_node\_with\_param,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set1\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

],,other\_handler2)

genElStr3([

1\_: fixed\_: result\_set2,

2\_: assign\_: const\_: pos\_: arc,

3\_: fixed\_: node\_album

],other\_album2,)

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

label(not\_other\_album2)

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

label(output)

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

searchSetStr3([

1\_: fixed\_: result\_set,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: fixed\_: output\_set

])

//#Проверим является ли множество result\_set1 пустым

searchElStr3([

1\_: fixed\_: result\_set1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node

],,out)

selectYStr3([

1\_: fixed\_: result\_set1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: output\_set

])

//#Проверим является ли множество result\_set2 пустым

searchElStr3([

1\_: fixed\_: result\_set2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node

],,out)

selectYStr3([

1\_: fixed\_: result\_set2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: output\_set

])

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

label(out)

ui\_sheet\_create([

1\_: assign\_: sheet,

2\_: fixed\_: sheet\_type,

3\_: fixed\_: sheet\_title,

4\_: fixed\_: layout

])

ui\_output([1\_: fixed\_: sheet, 2\_: fixed\_: output\_set])

return()

end

поиск альбома по типу записи (файл search\_by\_type.m4scp)

входные параметры: узел, содержащий тип записи;

выходные параметры: все альбомы с данным типом записи.

Пример диалога:

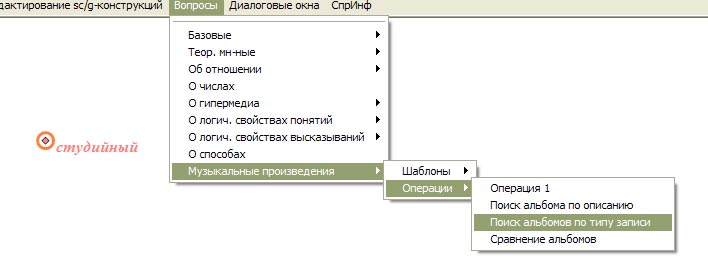


Рисунок 3.2.3 - вопрос «Поиск альбома по типу записи»



Рисунок 3.2.4 - ответ на вопрос «Поиск альбома по типу записи»

Алгоритм:

Делаем копию входных параметров.

Находим все узлы альбомов, подходящие под введенное описание и записываем их в множество 1.

Выводим на новый лист множество 1.

Исходный текст операции:

//#Операция поиска альбома по его описанию

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

procedure(search\_by\_type,

[[

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

sheet\_title = /"Операция прошла успешно"/;

atr1 = "/proc/keynode/1\_";

atr2 = "/proc/keynode/2\_";

ui\_information;

layout;

"MB\_TITLE\_QUERY\_RESULT" = /"Результат запроса"/;

"MB\_TEXT\_METKA" = /"Метка"/;

"MB\_TEXT\_ERROR1" = /"Не равны"/;

"MB\_TEXT\_ERROR2" = /"А сломалося!"/;

"Альбомы";

"описание\*";

"тип записи\*"

]],

[{

sheet,

shift,

node,

node1,

node\_temp,

node\_temp1,

arc,

arc1,

arc2,

attr\_arc,

selected\_els,

selected\_els\_copy,

result\_set,

temp\_node\_with\_param,

test\_node,

node\_param,

result,

set\_albums\_copy,

node\_album,

set\_album\_param,

temp\_node,

temp\_set\_1,

temp\_set\_2,

temp\_set\_3,

temp\_set\_5,

temp\_set\_0,

resultSet1,

resultSet2,

resultSet3,

resultSet4,

node\_out\_resSet2,

node\_out\_resSet3,

node\_out\_resSet4

}],

{[

1\_: in\_: selected\_els,

2\_: in\_: sheet,

3\_: in\_: shift

]}

)

genEl([

1\_: assign\_: node\_: result\_set

])

//#Копия selected\_els\_copy входных данных

searchSetStr3([

1\_: fixed\_: selected\_els,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: selected\_els\_copy

])

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: temp\_node\_with\_param

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Делаем копию set\_albums\_copy всех альбомов

searchSetStr3([

1\_: fixed\_: "Альбомы",

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: set\_albums\_copy

])

//#Проверяем описание каких альбомов удовлетворяет узел temp\_node\_with\_param

//#Берем один альбом node\_album из множества альбомов set\_albums\_copy

label(other\_album)

searchElStr3([

1\_: fixed\_: set\_albums\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: const\_: node\_album

],,not\_other\_album)

//#Удаляем дугу arc проведенную из set\_albums\_copy к node\_album

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#---------------2\_проходим тройку\_------------------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet3

])

searchSetStr3([

1\_: fixed\_: node\_out\_resSet3,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet4

])

//#Альбомы описываются по трем критериям, проверяем каждый из

//#---------------1\_Проверка по типу записи\_--------------

label(other\_handler)

searchElStr3([

1\_: fixed\_: resultSet4,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet4

],,other\_album)

//#Удаляем дугу arc проведенную из resultSet4 к node\_out\_resSet4

eraseEl([

1\_: fixed\_: arc

])

//#--------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_out\_resSet4,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "тип записи\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

],,other\_handler)

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#--------\_Выделяем одно из описаний типа записи node\_param\_----------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_param

])

//#--------\_Проверка равенства найденного описания и введенного пользователем\_-----

ifCoin([

1\_: fixed\_: node\_param,

2\_: fixed\_: temp\_node\_with\_param

],,other\_album)

//#--------\_Если альбом удовлетворяет условию, то заносим его в result\_set\_-------

genElStr3([

1\_: fixed\_: result\_set,

2\_: assign\_: const\_: pos\_: arc,

3\_: fixed\_: node\_album

],other\_album,other\_album,other\_album)

label(not\_other\_album)

ui\_sheet\_create([

1\_: assign\_: sheet,

2\_: fixed\_: sheet\_type,

3\_: fixed\_: sheet\_title,

4\_: fixed\_: layout

])

ui\_output([1\_: fixed\_: sheet, 2\_: fixed\_: result\_set])

return()

end

сравнение альбомов (файл ass.m4scp)

входные параметры: альбомы для сравнения;

выходные параметры: общие характеристики для сравниваемых альбом.

Пример диалога:

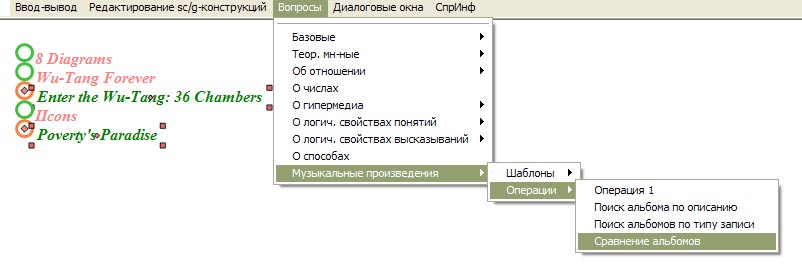


Рисунок 3.1.5 - вопрос «Поиск альбома по типу записи»

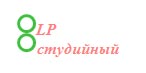


Рисунок 3.1.6 - ответ на вопрос «Поиск альбома по типу записи»

Алгоритм:

Делаем копию входных данных.

Находим все параметры описания для первого альбома и записываем в множество 1.

Находим все параметры описания для второго альбома и записываем в множество 2.

Находим пересечение множеств 1 и 2, записываем результат в множество 1.

Выводим на новый лист множество 1.

Исходный текст операции:

//#Операция общее между двумя альбомами

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

procedure(ass,

[[

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

sheet\_title = /"Операция прошла успешно"/;

atr1 = "/proc/keynode/1\_";

atr2 = "/proc/keynode/2\_";

ui\_information;

layout;

"MB\_TITLE\_QUERY\_RESULT" = /"Результат запроса"/;

"MB\_TEXT\_METKA" = /"Метка"/;

"MB\_TEXT\_ERROR1" = /"Не равны"/;

"MB\_TEXT\_ERROR2" = /"А сломалося!"/;

"Альбомы";

"описание\*";

"тип записи\*"

]],

[{

sheet,

shift,

node,

node1,

node\_temp,

node\_temp1,

arc,

arc1,

arc2,

attr\_arc,

selected\_els,

selected\_els\_copy,

result\_set,

result\_set1,

result\_set2,

temp\_node\_with\_param,

test\_node,

node\_param,

result,

set\_albums\_copy,

node\_album,

set\_album\_param,

temp\_node,

temp\_set\_1,

temp\_set\_2,

temp\_set\_3,

temp\_set\_5,

temp\_set\_0,

resultSet1,

resultSet2,

resultSet3,

resultSet4,

node\_out\_resSet2,

node\_out\_resSet3,

node\_out\_resSet4,

output\_set,

set1,

set2,

set3,

set4,

node\_in,

node\_to\_result\_set1,

node\_to\_result\_set2,

node\_out,

node\_to\_description,

node\_start\_descr,

set\_with\_node\_start\_descrp,

node\_to\_result,

set\_album1,

set\_album2

}],

{[

1\_: in\_: selected\_els,

2\_: in\_: sheet,

3\_: in\_: shift

]}

)

genEl([

1\_: assign\_: node\_: output\_set

])

genEl([

1\_: assign\_: node\_: result\_set

])

genEl([

1\_: assign\_: node\_: result\_set1

])

genEl([

1\_: assign\_: node\_: result\_set2

])

genEl([

1\_: assign\_: node\_: result

])

genEl([

1\_: assign\_: node\_: set\_album1

])

genEl([

1\_: assign\_: node\_: set\_album2

])

//#Копия selected\_els\_copy входных данных

searchSetStr3([

1\_: fixed\_: selected\_els,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: selected\_els\_copy

])

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: node\_album

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchElStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node\_to\_description,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

])

//#---------------2\_проходим тройку\_------------------

searchSetStr3([

1\_: fixed\_: node\_to\_description,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_start\_descr,

set3\_: assign\_: set\_with\_node\_start\_descrp

])

//#--------------\_Записываем все параметры в сеты для каждого альбома----------------

label(next\_descrp)

searchElStr3([

1\_: fixed\_: set\_with\_node\_start\_descrp,

2\_: assign\_: arc,

3\_: assign\_: node\_in

],,next\_album)

//#----\_ydalili dugu k yzly\_---

eraseEl([

1\_: fixed\_: arc

])

//#--------\_Пройдем пятерочку\_----

searchElStr5([

1\_: assign\_: node\_out,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_in,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_: atr1

])

searchElStr5([

1\_: fixed\_: node\_out,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node\_to\_result,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_: atr2

])

genElStr3([

1\_: fixed\_: set\_album1,

2\_: assign\_: arc\_: arc,

3\_: fixed\_: node\_to\_result

],next\_descrp,)

//#-----\_povtaryaem vse dlya vtorogo alboma\_--------

label(next\_album)

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: node\_album

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchElStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node\_to\_description,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

])

//#---------------2\_проходим тройку\_------------------

searchSetStr3([

1\_: fixed\_: node\_to\_description,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_start\_descr,

set3\_: assign\_: set\_with\_node\_start\_descrp

])

//#--------------\_Записываем все параметры в сеты для каждого альбома----------------

label(next\_descrp2)

searchElStr3([

1\_: fixed\_: set\_with\_node\_start\_descrp,

2\_: assign\_: arc,

3\_: assign\_: node\_in

],,out)

//#----\_ydalili dugu k yzly\_---

eraseEl([

1\_: fixed\_: arc

])

//#--------\_Пройдем пятерочку\_----

searchElStr5([

1\_: assign\_: node\_out,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_in,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_: atr1

])

searchElStr5([

1\_: fixed\_: node\_out,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node\_to\_result,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_: atr2

])

genElStr3([

1\_: fixed\_: set\_album2,

2\_: assign\_: arc\_: arc,

3\_: fixed\_: node\_to\_result

],next\_descrp2,)

label(out)

//#--------\_Naxodim obschie elementy\_--------

selectYStr3([

1\_: fixed\_: set\_album2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: set\_album1

])

//#--------\_Вывод\_-------

ui\_sheet\_create([

1\_: assign\_: sheet,

2\_: fixed\_: sheet\_type,

3\_: fixed\_: sheet\_title,

4\_: fixed\_: layout

])

ui\_output([1\_: fixed\_: sheet, 2\_: fixed\_: set\_album1])

return()

end

поиск года основания исполнителя (файл year\_of\_foundation.m4scp)

входные параметры: узел, содержащий название исполнителя;

выходные параметры: год основания исполнителя.

Пример диалога:

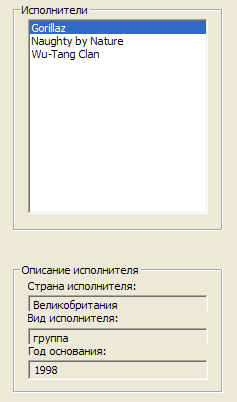


Рисунок 3.2.6 - вопрос год основания исполнителя

Алгоритм:

Делаем копию входных параметров.

Ищем год основания исполнителя.

Формируем результирующее множество.

Исходный текст операции:

//#Операция получения года основания выбранного исполнителя

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

procedure(year\_of\_foundation,

[[

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

sheet\_title = /"Операция прошла успешно"/;

atr1 = "/proc/keynode/1\_";

atr2 = "/proc/keynode/2\_";

ui\_information;

layout;

"MB\_TITLE\_QUERY\_RESULT" = /"Результат запроса"/;

"MB\_TEXT\_METKA" = /"Метка"/;

sall = "/etc/im\_keynode/get\_year";

cmd = "/operation/ui\_io\_exec\_command\_par/exec\_command\_par";

beg = /"...!!year\_of\_found\_performer!!..\n"/;

"описание\*";

"год основания\*";

"Исполнители"

]],

[{

sheet,

shift,

node,

node1,

node2,

node3,

node4,

node5,

performer,

arc,

arc1,

arc2,

arc3,

selected\_els,

selected\_els\_copy,

set,

set1,

set2,

set3,

set4,

result\_set,

output\_set,

out,

descr

}],

{[

1\_: in\_: selected\_els,

2\_: in\_: sheet,

3\_: in\_: shift

]}

)

//#Копия selected\_els\_copy входных данных

searchSetStr3([

1\_: fixed\_: selected\_els,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: selected\_els\_copy

])

label(other\_copy)

//#Выделяем один из элементов performer множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: performer

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к performer

eraseEl([

1\_: fixed\_: arc

])

//#Проходим первую пятерку пары

searchSetStr5([

1\_: assign\_: const\_: node1,

2\_: assign\_: const\_: arc1,

3\_: fixed\_: performer,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr1,

set1\_: assign\_: set

],,other\_copy,other\_copy)

//#Под отношением описание\*

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node,

set3\_: assign\_: set1

])

//#Находим пересечение множеств set и set1

selectYStr3([

1\_: fixed\_: set1,

2\_: assign\_: const\_: arc3,

3\_: assign\_: node\_ : node2,

set3\_: fixed\_: set

],,other\_copy,other\_copy)

searchElStr3([

1\_: fixed\_: set,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node3

])

//#Проходим вторую пятерку пары

searchSetStr5([

1\_: fixed\_: node3,

2\_: assign\_: const\_: arc1,

3\_: assign\_: node4,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr2,

set3\_: assign\_: set2

],,other\_copy,other\_copy)

searchElStr3([

1\_: fixed\_: set2,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node1

])

//#Проходим тройку

searchSetStr3([

1\_: fixed\_: node1,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node,

set3\_: assign\_: set

])

label(other\_node)

searchElStr3([

1\_: fixed\_: set,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node5

])

//#Удаляем дугу arc

eraseEl([

1\_: fixed\_: arc

])

//#Проходим первую пятерку второй пары

searchSetStr5([

1\_: assign\_: const\_: node1,

2\_: assign\_: const\_: arc1,

3\_: fixed\_: node5,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr1,

set1\_: assign\_: set3

],,other\_node,other\_node)

//#Под отношением год основания\*

searchSetStr3([

1\_: fixed\_: "год основания\*",

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node,

set3\_: assign\_: set4

])

//#Находим пересечение множеств set3 и set4

selectYStr3([

1\_: fixed\_: set4,

2\_: assign\_: const\_: arc3,

3\_: assign\_: node\_ : node2,

set3\_: fixed\_: set3

],,other\_node,other\_node)

searchElStr3([

1\_: fixed\_: set3,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node3

])

//#Проходим вторую пятерку второй пары

searchSetStr5([

1\_: fixed\_: node3,

2\_: assign\_: const\_: arc1,

3\_: assign\_: const\_: node5,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr2,

set3\_: assign\_: out

])

printNl([1\_: fixed\_: beg])

callReturn([1\_: fixed\_: cmd,

2\_: fixed\_:{[

1\_: out,

2\_: sheet,

3\_: sall

]}

], descr)

return()

end

поиск композиции по рейтингу (файл search\_by\_rating.m4scp)

входные параметры: узел, содержащий название исполнителя;

выходные параметры: год основания исполнителя.

Пример диалога:

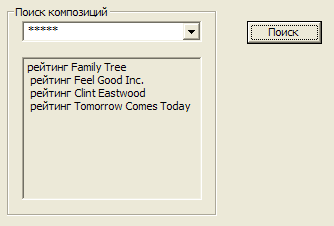


Рисунок 3.2.7 - вопрос композиции с выбранным рейтингом

Алгоритм:

Делаем копию входных параметров.

Ищем композиции по рейтингу.

Формируем результирующее множество.

Исходный текст операции:

//#Операция получения страны выбранного исполнителя

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

procedure(search\_by\_rating,

[[

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

sheet\_title = /"Операция прошла успешно"/;

atr1 = "/proc/keynode/1\_";

atr2 = "/proc/keynode/2\_";

ui\_information;

layout;

"MB\_TITLE\_QUERY\_RESULT" = /"Результат запроса"/;

"MB\_TEXT\_METKA" = /"Метка"/;

sall = "/etc/im\_keynode/get\_search\_rating";

cmd = "/operation/ui\_io\_exec\_command\_par/exec\_command\_par";

beg = /"...!!search\_by\_rating!!..\n"/;

"описание\*";

"место в чарте\*";

"Исполнители"

]],

[{

sheet,

shift,

node,

node1,

node2,

node3,

node4,

node5,

node6,

performer,

arc,

arc1,

arc2,

arc3,

selected\_els,

selected\_els\_copy,

output\_set,

set,

set1,

set2,

set3,

set4,

set\_bean,

out,

descr

}],

{[

1\_: in\_: selected\_els,

2\_: in\_: sheet,

3\_: in\_: shift

]}

)

genEl([

1\_: assign\_: node\_: out

])

//#Копия selected\_els\_copy входных данных

searchSetStr3([

1\_: fixed\_: selected\_els,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: selected\_els\_copy

])

label(other\_copy)

//#Выделяем один из элементов performer множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: performer

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к performer

eraseEl([

1\_: fixed\_: arc

])

//#Проходим первую пятерку пары

searchSetStr5([

1\_: assign\_: const\_: node1,

2\_: assign\_: const\_: arc1,

3\_: fixed\_: performer,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr2,

set1\_: assign\_: set\_bean

],,other\_copy,other\_copy)

//#Под отношением место в чарте\*

searchSetStr3([

1\_: fixed\_: "место в чарте\*",

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node,

set3\_: assign\_: set1

])

//#Находим пересечение множеств set и set1

selectYStr3([

1\_: fixed\_: set1,

2\_: assign\_: const\_: arc3,

3\_: assign\_: node\_ : node2,

set3\_: fixed\_: set\_bean

],,other\_copy,other\_copy)

label(other\_bean)

//#Выделяем один из элементов

searchElStr3([

1\_: fixed\_: set\_bean,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node6

],,to\_end,to\_end)

//#Удаляем дугу arc проведенную

eraseEl([

1\_: fixed\_: arc

])

searchSetStr5([

1\_: fixed\_: node6,

2\_: assign\_: const\_: arc1,

3\_: assign\_: const\_: node5,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr1,

set3\_: fixed\_: out

],other\_bean)

label(to\_end)

printNl([1\_: fixed\_: beg])

callReturn([1\_: fixed\_: cmd,

2\_: fixed\_:{[

1\_: out,

2\_: sheet,

3\_: sall

]}

], descr)

return()

end

поиск композиции по рейтингу (файл search\_by\_rating.m4scp)

входные параметры: узел, содержащий название альбома;

выходные параметры: обложка альбома.

Пример диалога:

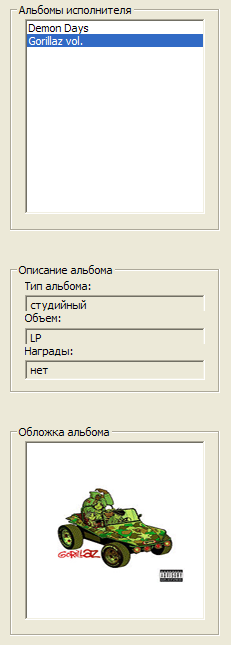


Рисунок 3.2.8 - вопрос композиции с выбранным рейтингом

Алгоритм:

Делаем копию входных параметров.

Ищем композиции по рейтингу.

Формируем результирующее множество.

Исходный текст операции:

//#Операция поиска альбома по его описанию

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

procedure(search\_by\_param,

[[

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

sheet\_title = /"Операция прошла успешно"/;

atr1 = "/proc/keynode/1\_";

atr2 = "/proc/keynode/2\_";

ui\_information;

layout;

"MB\_TITLE\_QUERY\_RESULT" = /"Результат запроса"/;

"MB\_TEXT\_METKA" = /"Метка"/;

"MB\_TEXT\_ERROR1" = /"Не равны"/;

"MB\_TEXT\_ERROR2" = /"А сломалося!"/;

"Альбомы";

"описание\*";

"тип записи\*"

]],

[{

sheet,

shift,

node,

node1,

node\_temp,

node\_temp1,

arc,

arc1,

arc2,

attr\_arc,

selected\_els,

selected\_els\_copy,

result\_set,

result\_set1,

result\_set2,

temp\_node\_with\_param,

test\_node,

node\_param,

result,

set\_albums\_copy,

node\_album,

set\_album\_param,

temp\_node,

temp\_set\_1,

temp\_set\_2,

temp\_set\_3,

temp\_set\_5,

temp\_set\_0,

resultSet1,

resultSet2,

resultSet3,

resultSet4,

node\_out\_resSet2,

node\_out\_resSet3,

node\_out\_resSet4,

output\_set

}],

{[

1\_: in\_: selected\_els,

2\_: in\_: sheet,

3\_: in\_: shift

]}

)

genEl([

1\_: assign\_: node\_: output\_set

])

genEl([

1\_: assign\_: node\_: result\_set

])

genEl([

1\_: assign\_: node\_: result\_set1

])

genEl([

1\_: assign\_: node\_: result\_set2

])

//#Копия selected\_els\_copy входных данных

searchSetStr3([

1\_: fixed\_: selected\_els,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: selected\_els\_copy

])

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: temp\_node\_with\_param

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Делаем копию set\_albums\_copy всех альбомов

searchSetStr3([

1\_: fixed\_: "Альбомы",

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: set\_albums\_copy

])

//#Проверяем описание каких альбомов удовлетворяет узел temp\_node\_with\_param

//#Берем один альбом node\_album из множества альбомов set\_albums\_copy

label(other\_album)

searchElStr3([

1\_: fixed\_: set\_albums\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: const\_: node\_album

],,not\_other\_album)

//#Удаляем дугу arc проведенную из set\_albums\_copy к node\_album

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#---------------2\_проходим тройку\_------------------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet3

])

searchSetStr3([

1\_: fixed\_: node\_out\_resSet3,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet4

])

//#Альбомы описываются по трем критериям, проверяем каждый из

//#---------------1\_Проверка по типу записи\_--------------

label(other\_handler)

searchElStr3([

1\_: fixed\_: resultSet4,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet4

],,other\_album)

//#Удаляем дугу arc проведенную из resultSet4 к node\_out\_resSet4

eraseEl([

1\_: fixed\_: arc

])

//#--------Ищем альбом по его введенному описанию---------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_out\_resSet4,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: temp\_node\_with\_param,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set1\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

],,other\_handler)

genElStr3([

1\_: fixed\_: result\_set,

2\_: assign\_: const\_: pos\_: arc,

3\_: fixed\_: node\_album

],other\_album,)

label(not\_other\_album)

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

////

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: temp\_node\_with\_param

],,output)

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Делаем копию set\_albums\_copy всех альбомов

searchSetStr3([

1\_: fixed\_: "Альбомы",

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: set\_albums\_copy

])

//#Проверяем описание каких альбомов удовлетворяет узел temp\_node\_with\_param

//#Берем один альбом node\_album из множества альбомов set\_albums\_copy

label(other\_album1)

searchElStr3([

1\_: fixed\_: set\_albums\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: const\_: node\_album

],,not\_other\_album1)

//#Удаляем дугу arc проведенную из set\_albums\_copy к node\_album

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#---------------2\_проходим тройку\_------------------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet3

])

searchSetStr3([

1\_: fixed\_: node\_out\_resSet3,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet4

])

//#Альбомы описываются по трем критериям, проверяем каждый из

//#---------------1\_Проверка по типу записи\_--------------

label(other\_handler1)

searchElStr3([

1\_: fixed\_: resultSet4,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet4

],,other\_album1)

//#Удаляем дугу arc проведенную из resultSet4 к node\_out\_resSet4

eraseEl([

1\_: fixed\_: arc

])

//#--------Ищем альбом по его введенному описанию---------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_out\_resSet4,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: temp\_node\_with\_param,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set1\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

],,other\_handler1)

genElStr3([

1\_: fixed\_: result\_set1,

2\_: assign\_: const\_: pos\_: arc,

3\_: fixed\_: node\_album

],other\_album1,)

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

label(not\_other\_album1)

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

//

//#Выделяем один из элементов temp\_node\_with\_param множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: temp\_node\_with\_param

],,output)

//#Удаляем дугу arc проведенную из selected\_els\_copy к temp\_node

eraseEl([

1\_: fixed\_: arc

])

//#Делаем копию set\_albums\_copy всех альбомов

searchSetStr3([

1\_: fixed\_: "Альбомы",

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: set\_albums\_copy

])

//#Проверяем описание каких альбомов удовлетворяет узел temp\_node\_with\_param

//#Берем один альбом node\_album из множества альбомов set\_albums\_copy

label(other\_album2)

searchElStr3([

1\_: fixed\_: set\_albums\_copy,

2\_: assign\_: arc\_: pos\_: const\_: arc,

3\_: assign\_: node\_: const\_: node\_album

],,not\_other\_album2)

//#Удаляем дугу arc проведенную из set\_albums\_copy к node\_album

eraseEl([

1\_: fixed\_: arc

])

//#Найдем множество set\_album\_param всех описаний альбома node\_album

//#---------------1\_проходим первую пару\_------------------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_album,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

])

searchElStr3([

1\_: fixed\_: resultSet2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet2

])

searchSetStr5([

1\_: fixed\_: node\_out\_resSet2,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: const\_: node\_: node,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set3\_: assign\_: resultSet3

])

//#---------------2\_проходим тройку\_------------------

searchElStr3([

1\_: fixed\_: resultSet3,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet3

])

searchSetStr3([

1\_: fixed\_: node\_out\_resSet3,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: assign\_: node,

set3\_: assign\_: resultSet4

])

//#Альбомы описываются по трем критериям, проверяем каждый из

//#---------------1\_Проверка по типу записи\_--------------

label(other\_handler2)

searchElStr3([

1\_: fixed\_: resultSet4,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node\_out\_resSet4

],,other\_album2)

//#Удаляем дугу arc проведенную из resultSet4 к node\_out\_resSet4

eraseEl([

1\_: fixed\_: arc

])

//#--------Ищем альбом по его введенному описанию---------

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: node\_out\_resSet4,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr1,

set1\_: assign\_: resultSet1

])

searchSetStr5([

1\_: assign\_: const\_: node\_: node,

2\_: assign\_: pos\_: arc\_: const\_: arc,

3\_: fixed\_: temp\_node\_with\_param,

4\_: assign\_: pos\_: arc\_: const\_: arc1,

5\_: fixed\_:atr2,

set1\_: assign\_: resultSet2

])

selectYStr3([

1\_: fixed\_: resultSet1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: resultSet2

],,other\_handler2)

genElStr3([

1\_: fixed\_: result\_set2,

2\_: assign\_: const\_: pos\_: arc,

3\_: fixed\_: node\_album

],other\_album2,)

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

label(not\_other\_album2)

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

label(output)

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

searchSetStr3([

1\_: fixed\_: result\_set,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: fixed\_: output\_set

])

//#Проверим является ли множество result\_set1 пустым

searchElStr3([

1\_: fixed\_: result\_set1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node

],,out)

selectYStr3([

1\_: fixed\_: result\_set1,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: output\_set

])

//#Проверим является ли множество result\_set2 пустым

searchElStr3([

1\_: fixed\_: result\_set2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node

],,out)

selectYStr3([

1\_: fixed\_: result\_set2,

2\_: assign\_: const\_: pos\_: arc,

3\_: assign\_: node\_ : node,

set3\_: fixed\_: output\_set

])

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

label(out)

ui\_sheet\_create([

1\_: assign\_: sheet,

2\_: fixed\_: sheet\_type,

3\_: fixed\_: sheet\_title,

4\_: fixed\_: layout

])

ui\_output([1\_: fixed\_: sheet, 2\_: fixed\_: output\_set])

return()

end

поиск композиции по рейтингу (файл type\_rec.m4scp)

входные параметры: узел, содержащий название альбома;

выходные параметры: тип альбома.

Пример диалога:

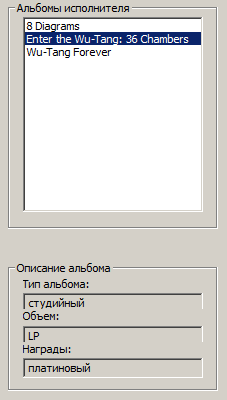


Рисунок 3.2.9 - вопрос тип выбранного альбома

Алгоритм:

Делаем копию входных параметров.

Ищем тип записи альбома.

Формируем результирующее множество.

Исходный текст операции:

//#Операция получения типа записи альбома

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

procedure(type\_rec,

[[

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

sheet\_title = /"Операция прошла успешно"/;

atr1 = "/proc/keynode/1\_";

atr2 = "/proc/keynode/2\_";

ui\_information;

layout;

"MB\_TITLE\_QUERY\_RESULT" = /"Результат запроса"/;

"MB\_TEXT\_METKA" = /"Метка"/;

sall = "/etc/im\_keynode/get\_type";

cmd = "/operation/ui\_io\_exec\_command\_par/exec\_command\_par";

beg = /"...!!type\_of\_track!!..\n"/;

"описание\*";

"тип записи\*";

"Исполнители"

]],

[{

sheet,

shift,

node,

node1,

node2,

node3,

node4,

node5,

performer,

arc,

arc1,

arc2,

arc3,

selected\_els,

selected\_els\_copy,

set,

set1,

set2,

set3,

set4,

result\_set,

output\_set,

out,

descr

}],

{[

1\_: in\_: selected\_els,

2\_: in\_: sheet,

3\_: in\_: shift

]}

)

//#Копия selected\_els\_copy входных данных

searchSetStr3([

1\_: fixed\_: selected\_els,

2\_: assign\_: const\_: pos\_: arc\_: arc,

3\_: assign\_: node\_: const\_: node,

set3\_: assign\_: selected\_els\_copy

])

label(other\_copy)

//#Выделяем один из элементов performer множества selected\_els\_copy

searchElStr3([

1\_: fixed\_: selected\_els\_copy,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: performer

])

//#Удаляем дугу arc проведенную из selected\_els\_copy к performer

eraseEl([

1\_: fixed\_: arc

])

//#Проходим первую пятерку пары

searchSetStr5([

1\_: assign\_: const\_: node1,

2\_: assign\_: const\_: arc1,

3\_: fixed\_: performer,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr1,

set1\_: assign\_: set

],,other\_copy,other\_copy)

//#Под отношением описание\*

searchSetStr3([

1\_: fixed\_: "описание\*",

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node,

set3\_: assign\_: set1

])

//#Находим пересечение множеств set и set1

selectYStr3([

1\_: fixed\_: set1,

2\_: assign\_: const\_: arc3,

3\_: assign\_: node\_ : node2,

set3\_: fixed\_: set

],,other\_copy,other\_copy)

searchElStr3([

1\_: fixed\_: set,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node3

])

//#Проходим вторую пятерку пары

searchSetStr5([

1\_: fixed\_: node3,

2\_: assign\_: const\_: arc1,

3\_: assign\_: node4,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr2,

set3\_: assign\_: set2

],,other\_copy,other\_copy)

searchElStr3([

1\_: fixed\_: set2,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node1

])

//#Проходим тройку

searchSetStr3([

1\_: fixed\_: node1,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node,

set3\_: assign\_: set

])

label(other\_node)

searchElStr3([

1\_: fixed\_: set,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node5

])

//#Удаляем дугу arc

eraseEl([

1\_: fixed\_: arc

])

//#Проходим первую пятерку второй пары

searchSetStr5([

1\_: assign\_: const\_: node1,

2\_: assign\_: const\_: arc1,

3\_: fixed\_: node5,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr1,

set1\_: assign\_: set3

],,other\_node,other\_node)

//#Под отношением тип записи\*

searchSetStr3([

1\_: fixed\_: "тип записи\*",

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node,

set3\_: assign\_: set4

])

//#Находим пересечение множеств set3 и set4

selectYStr3([

1\_: fixed\_: set4,

2\_: assign\_: const\_: arc3,

3\_: assign\_: node\_ : node2,

set3\_: fixed\_: set3

],,other\_node,other\_node)

searchElStr3([

1\_: fixed\_: set3,

2\_: assign\_: const\_: arc,

3\_: assign\_: const\_: node3

])

//#Проходим вторую пятерку второй пары

searchSetStr5([

1\_: fixed\_: node3,

2\_: assign\_: const\_: arc1,

3\_: assign\_: const\_: node5,

4\_: assign\_: const\_: arc2,

5\_: fixed\_: atr2,

set3\_: assign\_: out

])

printNl([1\_: fixed\_: beg])

callReturn([1\_: fixed\_: cmd,

2\_: fixed\_:{[

1\_: out,

2\_: sheet,

3\_: sall

]}

], descr)

return()

end

3.3 Верификация и отладка программ специфицированных операций

Отладка производилась с помощью Notepad++, SRS Studio и сообщений в командной строке. В процессе отладки текущей версии системы был выявлен следующий ряд ошибок:

1. Добавление в файл menu.scsy нового пункта меню: SRS Studio запускалась, но меню было не доступно .

Типовые ошибки:

* стояли лишние запятые в тексте программы;
* не хватало запятых;
* не хватало закрывающих list\_next скобок.

2. Добавление nsm-комманд в файл event.scsx: при попытке нажать на пункт меню, соответствующий nsm-комманде выводилось сообщение «Ошибка!Описание данного события не найдено!»

Ошибка: в файле event.scsx указан неправильный id или ссылка на операцию.

Типовые ошибки, допущенные в процессе написания SCP-программы:

* не объявлены используемые переменные;
* поставлены лишние запятые;
* не проставлены необходимые запятые;
* проставлены лишние пробелы (при копировании операторов из Руководства по SCP v0.7);
* проставлены атрибуты fixed\_ вместо assign\_ и наоборот.

В результате таких ошибок при выборе соответствующего пункта меню ничего не происходило.

3.4 Спецификация разработанных операций и библиотеки программ, сертификация разработанных ip-компонентов

В процессе разработки был реализовано более 20 различных scp-операций поиска. Выделить среди них какой-либо ip-компонент, на данный момент, не представляется возможным, т.к. тема проекта довольно специализированная и интеграции с другими системами не призводилось.

4. Пользовательский интерфейс проектируемой интеллектуальной справочной системы

4.1 Задачно-ориентированная спецификация пользовательского интерфейса проектируемой интеллектуальной системы

4.1.1 Декомпозиция пользовательского интерфейса

Пользовательский интерфейс данной интеллектуальной справочной системы можно разделить на ряд подсистем в зависимости от задачи, на которую они ориентированы:

Подсистема диалога с пользователем (windows-окна, scs-окна, диалоговые окна, элементы управления);

Подсистема трансляции с языка SC на естественный язык;

Подсистема обработки действий пользователя;

Подсистема хранения данных для отображения на экране.

4.1.2 Список используемых ip-компонентов пользовательского интерфейса

Для разработки данной интеллектуальной справочной системы были использованы следующие ip-компоненты:

Окна scs-интерфейса

диалоговые scs-окна

Окно раздела справочной информации SRS

это srs-окно,

класс sheet

класс shell

атомарные системные интерфейсные команды:

main\_menu

update\_menu

reduce\_menu

load\_folder

load\_scg

save\_scg

build\_srs

erase\_set

open\_gwf

change\_colour

reduction

increase

select\_all

modify\_scg\_el\_content

modify\_scg\_el\_type

modify\_scg\_el\_name

erase\_scg\_countur

erase\_scg\_el

paste

copy

gen\_srswindow

child\_command

4.2 Проектирование интерфейсных подсистем

Интерфейсная подсистема является интеллектуальной системой, построенной по семантическим технологиям. Поэтому для каждой интерфейсной подсистемы проектируется своя база знаний и машина обработки знаний.

База знаний пользовательского интерфейса интеллектуальной справочной системы по МП включает в себя ключевые узлы интерфейсных команд (файл em\_keynode), описание обработчиков интерфейсных команд(event.scsy), дерево событий(event.scsy).

Машина обработка знаний пользовательского интерфейса включает в себя следующие классы интерфейсных операций:

• рецепторные операции(пользователь-память)

• системные эффекторные операции(память-память)

• пользовательские эффекторные операции (память-пользователь)

Примером рецепторной операции является функция обработки нажатия кнопки, она осуществляет генерацию множества выходных параметров:

void pluginView::OnBnClickedButton2()

{

CComboBox \*combo = new CComboBox();

combo = (CComboBox\*)GetDlgItem(IDC\_COMBO1);

CString str;

int index = combo->GetCurSel();

combo->GetLBText(index, str);

m\_pSheet->sendMyEv("323322222222222221", str);//получить композиции с таким рейтингом

}

Примером системных эффекторных операций является операция ui\_io\_exec\_command, которая генерирует результирующую sc-конструкцию интерфейсной операции для последующей её обработки:

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

//

//

#include "scp\_keynodes.scsy"

#include "../seb/planimetry.scsy"

#include "im\_keynodes.scsy"

procedure(exec\_command,

[[

"main window"= "/etc/im\_keynode/main window";

ch\_comm = "/etc/im\_keynode/child\_command";

$errors\_found = /"Неверно заданы аргументы."/;

sheet\_type = "/proc/agents/shell/keynode/SCgSheet";

layout =c= /"logical"/;

p\_sheet\_title =c= /"Результаты верификации"/;

1\_, 2\_;

]],

[{

notfound,main\_wnd,show\_set,comm,p2,s,cmd,

arc,a,arc2,

resultall,

command,

segment,scg\_el,wnd,sheet,set\_scg\_el

}],

{[

1\_: in\_: set\_scg\_el,

2\_: in\_: sheet,

3\_: in\_: cmd

]}

)

genEl([

1\_: assign\_: node\_:const\_:resultall

])

searchElStr3([1\_: fixed\_: "main window",

2\_: assign\_: a,

3\_: assign\_: main\_wnd],,notfound)

sys\_get\_location([1\_: fixed\_: main\_wnd, 2\_: assign\_: segment])

sys\_set\_default\_segment([1\_: fixed\_: segment])

genEl([

1\_: assign\_: node\_:const\_:command

])

genElStr3([

1\_: fixed\_: ch\_comm,

2\_: assign\_: arc\_:const\_: pos\_: arc,

3\_: fixed\_: command

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_:const\_: pos\_: a,

3\_: fixed\_: ch\_comm

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_:const\_: pos\_: a,

3\_: fixed\_: command

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_:const\_: pos\_: a,

3\_: fixed\_: arc

])

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

// создание параметров команды

//

// первый параметр

genElStr5([1\_: fixed\_: command,

2\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc,

3\_: fixed\_: sheet,

4\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc2,

5\_: fixed\_: 1\_

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_:const\_: pos\_: a,

3\_: fixed\_: sheet

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_:const\_: pos\_: a,

3\_: fixed\_: 1\_

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_:const\_: pos\_: a,

3\_: fixed\_: arc

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_:const\_: pos\_: a,

3\_: fixed\_: arc2

])

// генерация узла связки второго параметра команды

genElStr5([1\_: fixed\_: command,

2\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc,

3\_: assign\_: const\_: node\_: p2,

4\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc2,

5\_: fixed\_: 2\_

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: p2

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: 2\_

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc2

])

// генерация второго параметра

genElStr5([1\_: fixed\_: p2,

2\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc,

3\_: assign\_: const\_: node\_: s,

4\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc2,

5\_: fixed\_: 1\_

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: s

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc2

])

// формируем дочернюю команду

genElStr5([1\_: fixed\_: cmd,

2\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc,

3\_: assign\_: const\_: node\_: comm,

4\_: assign\_: const\_: actual\_: pos\_: temporary\_: arc2,

5\_: fixed\_: s

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: comm

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc2

])

genElStr3([

1\_: fixed\_: s,

2\_: assign\_: arc\_: const\_: pos\_: arc,

3\_: fixed\_: comm

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc

])

genElStr3([

1\_: fixed\_: s,

2\_: assign\_: arc\_: const\_: pos\_: arc,

3\_: fixed\_: cmd

])

genElStr3([

1\_: fixed\_: resultall,

2\_: assign\_: arc\_: const\_: pos\_: a,

3\_: fixed\_: arc

])

varAssign([1\_:assign\_:wnd,2\_:fixed\_:main\_wnd],ok\_msg,ok\_msg)

// если произошла ошибочная ситуация

label(err\_msg)

genElStr3([1\_: fixed\_:resultall,

2\_: assign\_: const\_: temporary\_: actual\_: pos\_: a,

3\_: fixed\_: $errors\_found])

ui\_sheet\_create([ 1\_: assign\_: const\_: node\_: wnd,

2\_: fixed\_: sheet\_type,

3\_: fixed\_: p\_sheet\_title,

4\_: fixed\_: layout])

printEl([1\_: resultall])

// вывод результата

label(ok\_msg)

ui\_send([1\_: fixed\_: wnd, 2\_: fixed\_: resultall])

eraseEl([1\_:fixed\_: f\_: resultall])

return()

end

Примером пользовательской эффекторной операции (память-пользователь) является функция setAlbumList, которая обрабатывает ключевой узел интерфейсной команды для отображения идентификаторов выходных параметров в элемент управления:

void pluginSheet::setAlbumList(sc\_addr addr)

{

CArray<sc\_string, sc\_string&> albumList;

albumList.FreeExtra();

Shell\* shell = Shell::getInstance();

sc\_session\* s = shell->getSession();

sc\_iterator\* it = s->create\_iterator(sc\_constraint\_new(CONSTR\_3\_f\_a\_a, addr, 0, 0),true);

for(; !it->is\_over(); it->next())

{

sc\_addr album\_addr = it->value(2);

sc\_string str = s->get\_idtf(album\_addr);

albumList.Add(str);

}

m\_pView->UpdateListBox2(albumList);

}

4.3 Верификация и отладка пользовательского интерфейса

Для реализации и тестирования базы данных использовался SRS Studio. Для написания кода операций использовал Notepad++, интерфейс реализован по средствам библиотек MFC на языке программирования С++, верификация и тестирование с помощью SRS Studio, Microsoft Visual Studio .NET 2003 и лога pm в командной строке.

4.4 Спецификация разработанного пользовательского интерфейса, сертификация разработанных ip-компонентов

5. Интеграция разработанной системы с другими системами

Интеллектуальная справочная система может быть интегрирована с другими система, работающими в данной предметной области.

Кроме этого могут быть заимствованы операции и ip-компоненты для качественного улучшения системы.

Интеграция позволит расширить как базу знаний, так и перечень реализуемых операций, что сделает систему еще более востребованной для конечного пользователя.

Но вместе с тем необходимо оценить затраты необходимые для интеграции (приведение базы знаний к одинаковым ключевым элементам, использование одинаковых конструкций, соответствующая реализация определенных операций), и на основе анализа выбрать наилучшую стратегию.

С целью облегчения процесса интеграции необходимо разрабатывать ip-компоненты всех подсистем интеллектуальной системы (базы знаний, машины обработки знания и пользовательского интерфейса), которые не зависят от реализации конкретной базы знаний.

6. Направления дальнейшего развития прототипа интеллектуальной справочной системы и пути промышленного изготовления спроектированной интеллектуальной системы

Для полноценного функционирования справочной системы и достижения главной цели – способность системы отвечать на самые нетривиальные вопросы и привлекать широкие массы пользователей, нужно:

Дальнейшие развитие базы знаний, добавление новых предметных узлов и ключевых элементов. Кроме этого база должна пополняться и расширяться новыми знаниями, которые будут генерироваться при работе операций.

Развитие машины обработки заключается в увеличении количества классов задач, которые будет решать данная интеллектуальная справочная система.

Пользовательский интерфейс прикладной системы должен давать возможность работать как с информацией, представленной на языках, являющихся подмножествами языка SC, так и на естественном языке.

В первую очередь необходимо уделить большое внимание естественно-языковому интерфейсу, так как SCg-конструкции, которые система генерирует на данный момент, являются слишком громоздкими и не удобными для восприятия. Ответ на формальном языке представляет собой дополнительный барьер для понимания информации и делает систему не пригодной для пользователей, не знакомых с теорией множеств и языком SC.

Пользовательский интерфейс должен давать возможность пользователю сосредоточиться на постановке задачи, а не на способе её достижения. Время и усилия, которые необходимы пользователю для достижения цели, должны сводиться к минимуму.

Следует уделить внимание «дружелюбности» интерфейса. Необходимо стремиться к тому, чтобы даже не опытному пользователю было понятно, как работать с системой. Время, которое необходимо для освоения навыков работы со средой интеллектуальной справочной системы должно сводиться к минимуму.

7. Вклад в развитие семантических технологий проектирования интеллектуальных систем

7.1 Список выявленных ошибок среды проектирования интеллектуальных систем

После закрытия всех окон SRS-Studio, в системе остаётся незавершенный процесс, который не даёт запуститься собранной системе или ещё одной копии SRS-Studio.

При загрузке sc-элементов в память не загружаются дуги.

7.2 Список предложений по совершенствованию среды проектирования интеллектуальных систем

Доработка руководства пользователя по операциям на языке SCP.

Разработка инструментария для написания scp-программ (минимум отладчика).

Написание новых scp-функций (например, переход по паре).

Исправление указанных ошибок.

Создание ip-компонентов базы знаний, машины обработки знаний, пользовательского интерфейса.

Отладка текущего инструментария для уменьшения используемых ресурсов.

7.3 Список предложений по совершенствованию семантических технологий проектирования интеллектуальных систем

Необходима разработка инструментария качественно нового уровня (предложения по развитию представлены в разделе 7.3.), так как работа с инструментарием на данном этапе затрудняет использование данной технологии.

Изучение в рамках учебного процесса студентами теоретических основ, необходимых для использования и внедрения данной технологии (теория множеств, модели представления знаний, язык программирования scp и др.)

Так же важным этапом в развитии технологии является создание ip-компонентов, задача которых упростить и ускорить процесс разработка за счёт повторного использования каких-либо элементов интеллектуальной системы. Важным является создание ip-компонентов всех подсистем: базы знаний, машины обработки знаний, пользовательского интерфейса.

7.4 Создание и использование ip-компонентов

Созданная БЗ, набор операций и отдельные элементы пользовательского интерфейса могут быть использованы в интеллектуальных справочных системах смежных областей, в качестве ip-компонентов. Что позволит ускорить их разработку, расширить функционал. Таким образом с бесконечным числом интеграций с другими интеллектуальными справочными системами, получится система, которая будет хранить бесконечное количество информации, отвечать на любые поставленные нетривиальные вопросы, а общение между пользователем и системой будет происходить посредством дружелюбного пользовательского интерфейса.

Библиографический указатель

1. Режим доступа: http://www.skilfully.org/18.08.2008/1. - Дата доступа: 21.12.2008.
2. Режим доступа: http://www.encyclopedia.ru/news/enc/detail/2321/. – Дата доступа: 21.12.2008.
3. Справочно-информационный портал Википедия. - Режим доступа: http://ru.wikipedia.org/wiki/Семантическая\_паутина. - Дата доступа: 20.12.2008.
4. The Semantic Web. Scientific American, 17 мая 2001, русский перевод: Семантическая Сеть.
5. Справочно-информационный портал Википедия. - Режим доступа: http://ru.wikipedia.org/wiki/Семантическая\_сеть. - Дата доступа: 21.12.2008.
6. http://www.trizland.ru/trizba.php?id=186. – Дата доступа: 21.12.2008.
7. Руководство разработчика семантической справочной системы, Минск, 2008.
8. Руководство по программированию на языке SCP, Минск, 2005.