# 1) Введение

В настоящее время предметом исследований большинства современных разработчиков информационных систем являются методы и подходы, чтобы построить не просто распределенные информационные системы, но и интеллектуальные, предусматривающие возможность их адаптации к конкретной предметной области. Подобная потребность появилась при рассмотрении распределенной системы отслеживания задач по технике Помодоро, чтобы исключить необходимость в роли человека, распределяющего задачи, и автоматизировать этот процесс оптимально.

Существует множество инструментов для исследований и разработки таких систем, анализа больших массивов данных с учетом специфики предметной области. Одной из наиболее универсальных методик представления экспертных знаний, в данном случае знаний при распределении заданий по исполнителям, полностью описывающие семантически, является онтологический подход.

Актуальность применения онтологий для реализации менеджмента описана в статье «Онтология для управления проектами: обследование» [[14](#_fbqlbsjopug9)], где представлен систематический обзор литературы, посвященный онтологиям управления проектами. Автор выявил отсутствие стандартизации в терминологии и концепциях, отсутствие систематического моделирования областей и использования онтологий, главным образом, в прототипных системах онтологий, которые затрагивают довольно ограниченные аспекты процессов управления.

Цель работы: разработать и исследовать систему распределения заданий по исполнителям с применением онтологии.

Задачи исследовательской работы:

* изучить существующие решения распределения в тайм-менеджменте;
* рассмотреть применение онтологии в управлении задачами и проектами;
* разработать собственную реализацию алгоритма распределения;
* оценить эффективность и применимость разработанного алгоритма.

Многие авторы представляют данную задачу как задачу системы поддержки принятия решений (СППР) и предлагают решить ее смешанным образом. Наиболее частое применение онтологического и прецедентного подходов.

В статье «Интеллектуальная поддержка принятия решений в организационном управлении разработкой программных проектов» [Л. Р. Черняховская, А. И. Малахова] [[1]](#footnote-1)рассматриваются проблемные ситуации при организационном управлении разработкой программных проектов, описание которых составляется на основе наиболее информативных характеристик управляемых процессов.

Поиск решений авторы осуществляют с использованием построенной на основе онтологического анализа нечеткой базы знаний, состоящей из:

* правил принятия решений на языке формализации онтологических правил SWRL (Semantic Web Rule Language) в проблемных ситуациях
* прецедентов решения конкретных проблемных ситуациях, которые затем преобразуются в правила поддержки принятия решений.

В базе знаний правила разделены на блоки, которые позволяют связать конкретный процесс с соответствующим набором правил в базе знаний, а также организовать иерархию правил и ускорить поиск необходимых решений с применением базы знаний.

Математическое обоснование формируемых решений производится с применением модуля оптимизации решений, взаимодействующего с модулем формирования решений и базой знаний через систему нечеткого вывода. Расчет оценок альтернатив в модуле оптимизации решений производится с использованием математических моделей, выбранных на основе онтологии поддержки принятия решений. В данной статье не описано, какие инструменты выбраны для реализации модулей.

Однако в качестве модуля оптимизации решений авторы другой статьи «Применение генетических алгоритмов в управлении программными проектами» [И.А.Гарматина, М.В.Маркитантов][[2]](#footnote-2) выбрали генетический метод оценки трудозатрат. Он позволяет более точно в сравнении с другими методами определить вектор движения того или определенного проекта по аналогии с эволюцией биологической системы и рассчитывать то дополнительное время, которое потребуется для устранения всех конфликтов при разработке ПО.

В работе «Онтология как основа семантического анализа данных в задачах автоматизированного проектирования» [В.С.Мошкин] для решения задач СППР слабоструктурированных информационных ресурсов так же предложено применение нечеткой онтологии (FuzzyOWL) и систем продукционных правил (SWRL). Такой способ используется совместно с механизмом прецедентов, которые «представляют собой простейшие логические структуры, связывающие условия со следствием»[[3]](#footnote-3). Для модуля логического вывода экспертных рекомендаций в порядке их релевантности авторы используют результаты анализа исходных параметров исследуемой системы и обработанной онтологии.

В статье «Система поддержки принятия решений в IT-подразделениях на основе интеграции прецедентного подхода и онтологии» [Авдеенко Т.В., Макарова Е.С.[[4]](#footnote-4)] рассматривалась область поддержки пользователей в IT-подразделениях, где ошибки пользователей консультантов-аналитиков описываются прецедентным подходом. В рамках ПрО статьи онтология описывает работу консультантов-аналитиков из трех таксономий: по бухгалтерскому учету, расчетам с персоналом, договорному блоку.

Авторы интегрируют прецеденты IT-консультирования с онтологиями предметной области и устанавливают ассоциативные связи прецедентов с концептами онтологии. Такой подход называется Case-Based Reasoning (CBR). Решение новых задач происходит из прошлых успешных прецедентов через их адаптацию к новым условиям. Когда рассматривается новая ситуация, система находит подобный прецедент в базе знаний как аналог решаемой задачи и пытается использовать решение найденного прецедента. Если необходимо, близкий прецедент адаптируется к текущей ситуации.

Терминалы (классы, не имеющие потомков) будут играть роль ключевых слов при индексировании прецедентов. Вес связи задается от концепта-родителя Ci(l-1) к концепту-ребенку Ci(l). Прецедент ассоциируется как терминальный концепт kwj (ключевое слово). Описанный алгоритм изображен на рис.1.1.

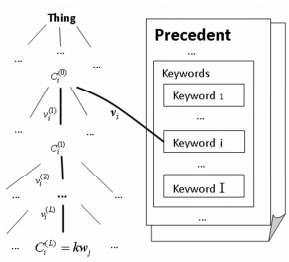


Рис. 1.1. Вычисление силы связи прецедента с ключевыми словами

Методы CBR уже применяются во множестве прикладных задач – в медицине, управлении проектами, для анализа и реорганизации среды, разработки товаров массового спроса с учетом предпочтений разных групп потребителей и т. д.

В работе «Методы добычи данных при построении локальной метрики в системах вывода по прецедентам» [Л. Е. Карпов[[5]](#footnote-5)] предложен метод принятия решений, основанный на совместном применении методов извлечения знаний и вывода по прецедентам. На момент написания статьи все существующие интегрированные системы подобного рода строят для себя модель данных как способ получения фонового знания. Меры близости определяют наиболее подходящих прецедентов и адаптации найденного решения. Для выбора прецедентов популярный метод "ближайшего соседа" авторы предлагают заменить на локальную контекстно-зависимую метрику. При ее построении может быть использована как предварительная кластеризация базы прецедентов, так и разбиение базы на классы эквивалентности с привлечением экспертного знания.

<вывод>

# 2) Формальная модель онтологии

Основными объектами проекта являются внутренние задачи IT-компании и сотрудники. Задачи будут иметь структуру:

Z = <D, T, P> при Z = {z1,... ,zn}, n–количество заданий

где

D – множество доменов, к которым относится задание (продукты компании, компоненты);

Т = {T0, Tend} – множество длительности исполнения задания, где

T0 – множество дат при постановке задания на исполнение,

Tend – множество дат после выполнения задания;

Р – множество оценок сложности задания.

Исполнители (assignee) будут иметь следующую структуру:

A = <D, T, P> при A = {a1,... ,am}, m–количество исполнителей,

где

D – множество области специализации (продукты компании, компоненты, аналогично);

Т = {T0, Tend} – множество длительности занятости исполнителя, где

T0 – множество дат начала занятости,

Tend – множество дат после выполнения задания;

Р – множество уровня квалификации.

# 3) Реализация системы распределения заданий на основе онтологического подхода

Основные объекты создаваемой системы и их признаки:

* внутренние задачи IT-компании:
  + область применения (домен),
  + длительность исполнения (время),
  + оценка сложности задания
* сотрудники компании:
  + области специализации (домен),
  + время занятости,
  + уровень квалификации.

Необходимо оптимально сопоставить признаки задач соответственно признакам сотрудников с применением онтологии.

Методы решаемых задач интеллектуального анализа:

* ограничения онтологии ПрО, позволяющие сузить выборку для распределения;
* задача классификации, непосредственно решающая задачу распределения;
* методы генетического алгоритма как альтернатива поиска оптимальных решений.

В рамках данной статьи рассматривается первый из описанных методов.

Данная система предназначается для предприятий IT области с большим объемом проектов. Она является частью системы управления задачами, ведет учет и распределение заданий на основе ограничений онтологии ПрО между пользователями, объединенными в группу.

Данные для онтологии используются из Административной БД сайта Redmine. БД сайта содержит задания по ошибкам (багам) во время разработки их веб-проекта.

## Представление онтологии в Protege

Источники с исходными данными будут, как формироваться вручную, так и извлекаться в виде csv, xml формата представления из открытой базы данных сайта Redmine (рис.3.1).

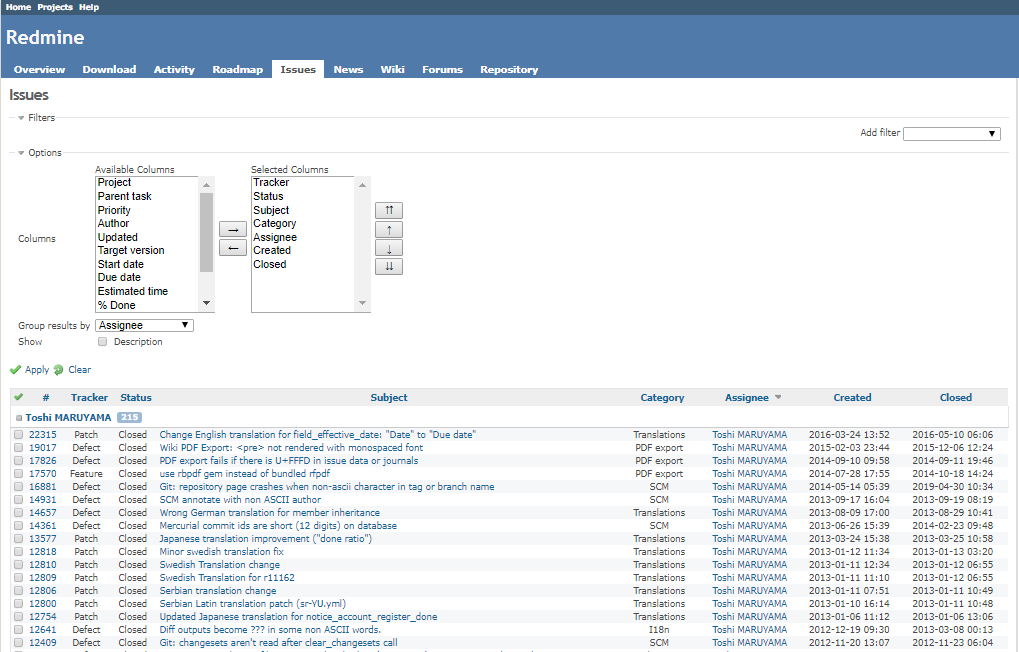


Рис.3.1. Страница отчетов сайта Redmine

Онтология задач IT-специалистов предполагает создание классов и добавлении ограничений в формате OWL. Онтология построена вручную (рис.3.2) с учетом малого опыта в использовании редакторов онтологий автором проекта.

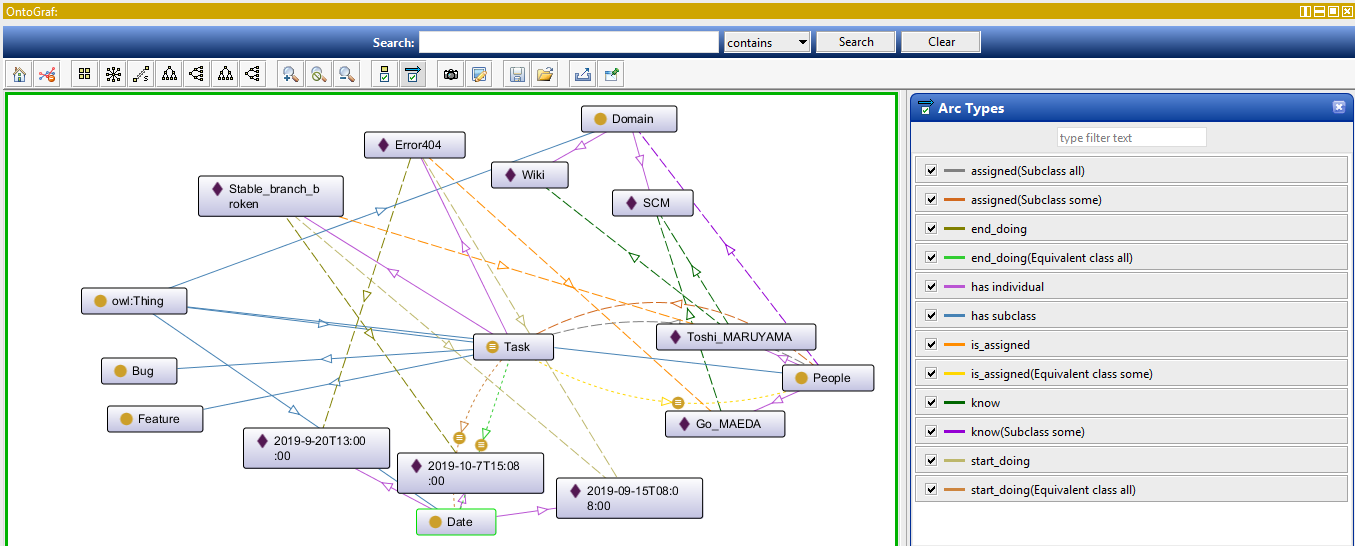


Рис.3.2. Онтология задач с указанием связей классов и индивидуумов

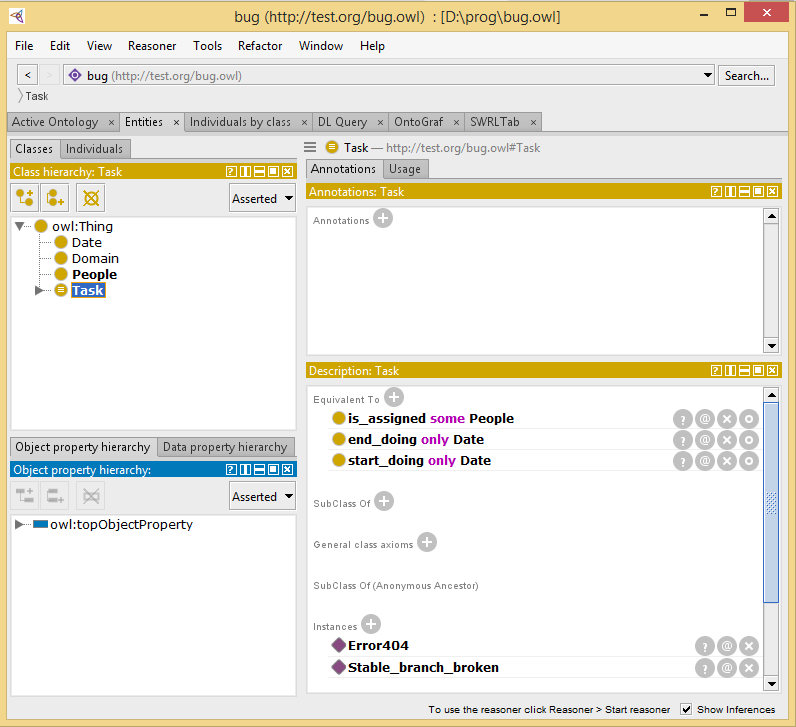


Рис.3.3 Описание класса Задач в онтологии

Онтология содержит классы с ограничениями кардинальности, что шире понятий домена и ранга (рис.3.3). Класс Задач (Task) имеет связи назначения (is assigned) Человеку (People), специализации (specialized) по Области применения (Domain), дату создания (start\_doing) по Дате (Date).

## Архитектура реализованной программной системы

Архитектура системы как модуль поведения описывается в нотации IDEF0. Такое функциональное описание бизнес-процессов представлено на рис.3.4. На данной диаграмме описан один из рассматриваемых способов распределения, методов классификации.

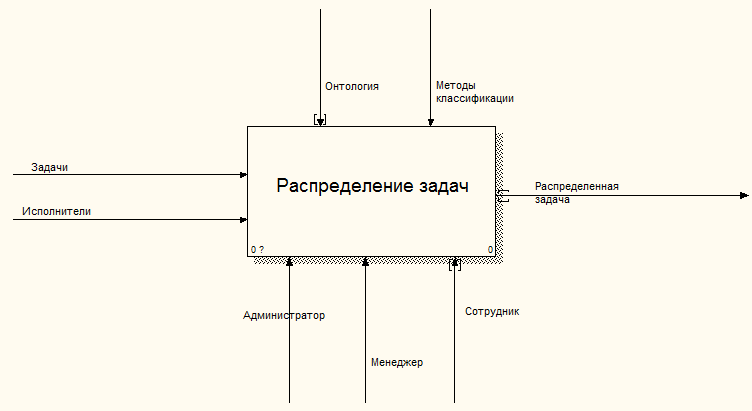


Рис.3.4. Диаграмма архитектуры системы

На вход получаем данные задачи и исполнителей. Они преобразуются с управляющими элементами: инструменты. Архитектура “черный ящик” снизу управляется пользователями. Подробнее рассмотрено на диаграмме последовательности (рис.3.5) отражает временные особенности передачи сообщений между частями программы, а также показывает время жизни классов.

В данной диаграмме частями программы являются основные модули программной разработки: главный модуль, объединяющий в себе работу модуля связи с онтологией и модуля классификации, оптимизирующего распределение. В рамках данного курсового проекта эта часть не будет реализована.

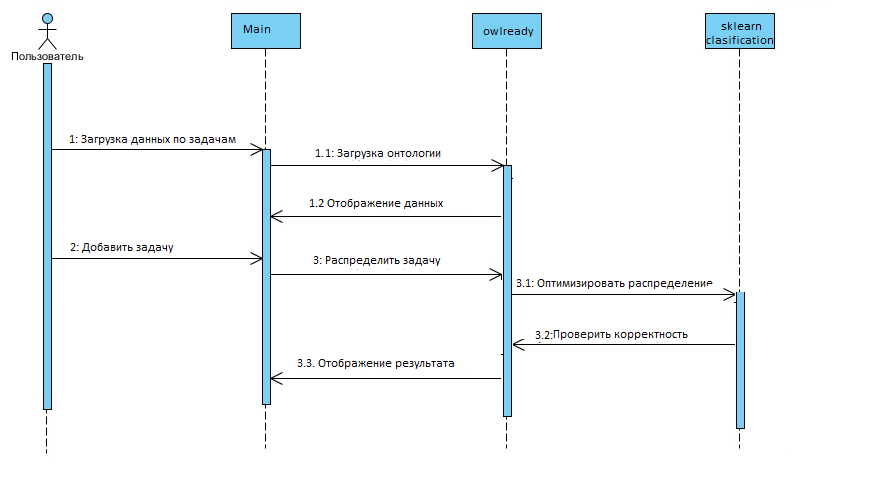


Рис.3.5 Диаграмма последовательности

# 4) Результаты вычислительных экспериментов;

Библиотека owlready [16] используется для работы с онтологией. Имеется официальное руководство обо всех ее возможностях с примером основных реализаций. Изначально создание и проектирование онтологии осуществлялось программным способом с помощью данной библиотеки. Данные загружаются автоматически по ссылке URI, имеющейся у каждой онтологии и представляются в табличном виде (рис.4.1).

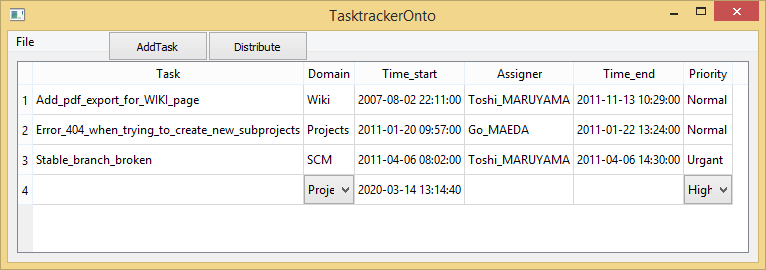


Рис.4.1. Окно системы распределения задач. Добавление задачи

Разработанная система сопоставляет активные задачи исполнителям с применением ограничения онтологии ПрО (рис.4.2).

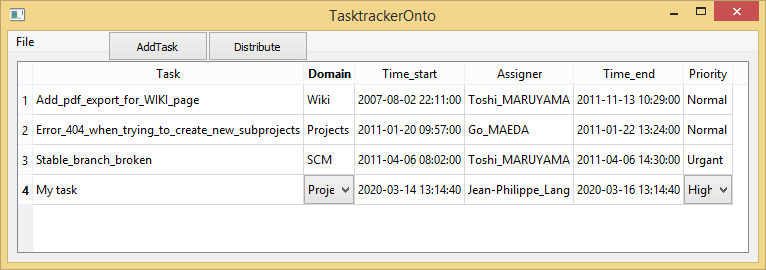


Рис.4.2.а Результат распределения новой задачи в окне системы.



Рис.4.2б. Окно консоли, отрабатывающее распределение новой задачи.

# 5) Заключение

Дополнительная формализация параметров задачи позволила управлять онтологией. Ограничения онтологии ПрО через связи и кардинальности сузили набор для распределения. Реализованные функции включают в себя простой старт и удобный интерфейс, визуализацию онтологии IT-задач в виде таблицы, отмеченных в аналогах.

Выбранная среда разработки python IDLE и множество поддерживаемых библиотек, ориентированных на интеллектуальный анализ, помогли реализовать desktop-приложение распределения задачи по группам исполнителей.

Для анализа качества работы распределения достаточно использовать дедукцию, а именно построить небольшую часть системы в онтологии, как это было сделано из ресурсов Redmine, и рассчитывать качество с изменением небольшого числа параметров.

При дальнейшем исследовании планируется экспериментировать методами классификации для выбора оптимального алгоритма. Так же будет использоваться менее формализованный язык, чем OWL. В высоко-формализованном представлении не оказалось смысла. Однако необходим так же повторный выбор инструментов для интеграции онтологии с программой.

# 6) Список литературы.

1. Вестник УГАТУ, 2013г [↑](#footnote-ref-1)
2. Студенческий: научный журнал. – № 6(6). Часть 1, Новосибирск: Изд. АНС «СибАК»,

   2017. – 80 с. (30-35 с.)– Электрон. версия. печ. публ. – <https://sibac.info/journal/student/6>. [↑](#footnote-ref-2)
3. Нечеткие системы и мягкие вычисления (НСМВ-2014): труды Шестой всероссийской научно – практической конференции. – Т.1. –СПб. : Политехникасервис, 2014. – С. 174-182. [↑](#footnote-ref-3)
4. Вестник АГТУ: Серия: Управление, вычислительная техника и информатика. 2017-№3 [↑](#footnote-ref-4)
5. Карпов Л.Е. Методы добычи данных при построении локальной метрики в системах вывода по прецедентам /Л. Е. Карпов, В. Н. Юдин//Препринты Института системного программирования РАН, Препринт 18, 2006 г. <http://citforum.ru/consulting/BI/data_mining/> [↑](#footnote-ref-5)