**Архитектура гибридной интеллектуальной** **информационной системы анализа судебной практики арбитражных судов**

Таран М.О., Гапанюк Ю.Е.

Таран Мария Олеговна – аспирантка МГТУ им. Н.Э.Баумана, Российская Федерация, г. Москва

Гапанюк Юрий Евгеньевич – к.т.н., доцент МГТУ им. Н.Э.Баумана, Российская Федерация, г. Москва

E-mail: gapyu@bmstu.ru

**Ключевые слова:** арбитражный суд, гибридная интеллектуальная система (ГИС), гибридная интеллектуальная информационная система (ГИИС), сознание информационной системы, подсознание информационной системы, анализ текстов, многоагентная система (МАС), граф, метаграф, метавершина, метаребро, метаграфовый процесс, метаграфовый агент, активный узел метаграфа.

**Аннотация**

**Цель работы:** совершенствование научно-методической базы информационного обеспечения системы судебной практики арбитражных судов.

**Методы:** методы проектирования гибридных интеллектуальных информационных систем на основе моделей сложных сетей.

**Результаты:** рассмотрен подход к разработке гибридных интеллектуальных информационных систем (ГИИС), основанный на использовании модулей сознания, подсознания и коммуникации. Рассмотрен подход на основе сложных сетей для реализации ГИИС, показано, что метаграфовая модель позволяет описывать данные, знания и процессы, как составные части ГИИС. Рассмотрена структура метаграфового агента, обеспечивающего обработку метаграфовой модели. Показано, что метаграфовый агент может быть представлен в виде метаграфовой модели, что позволяет агентам верхнего уровня модифицировать структуру агентов нижнего уровня. Введено понятие активного узла метаграфа как комбинации метаграфовой модели данных и знаний, и метаграфовых агентов. На основе активного узла метаграфа введено понятие метаграфового процесса. В результате предложена архитектура гибридной интеллектуальной информационной системы анализа судебной практики арбитражных судов.

**Введение**

Защита законных прав и интересов предпринимателей не обходится без взаимодействия с судебной системой. В арбитражные суды обращаются как индивидуальные предприниматели, так и юридические лица.

Судебный процесс состоит из этапов, каждый из которых сопровождается документальным оформлением. В завершении процесса суды принимают судебный акт, который может называться по-разному, в зависимости от инстанции и сути принятого решения. В соответствии с ч. 1 ст. 15 АПК РФ к судебным актам относят: судебный приказ, решение, постановление, определение.

Хотя судебная система России не является прецедентной, судебные решения занимают ключевую роль при подготовке к судебному процессу. Из-за законодательных коллизий или отсутствия регулирующих норм большинство судебных решений основывается на позициях Верховного Суда РФ и/или Высшего Арбитражного Суда РФ (был упразднен в 2014 году).

Представители как истцов, так и ответчиков в обязательном порядке начинают подготовку к делу с изучения судебной практики, поиска похожих дел, определения схожих обстоятельств и обоснований своих позиций. Мотивированный судебный акт может занимать от 4 до 15 страниц. При этом таких решений нужно изучить достаточно много, чтобы составить более полное представление о возможном исходе судебного процесса.

По информации сайта «Электронное правосудие», за девять месяцев 2019 года в арбитражных судах России было выпущено более 1 300 000 документов[[1]](#footnote-1). Обработать, обобщить и сделать какие-либо выводы из такого массива документов вручную достаточно сложно.

В соответствии с ч. 4 ст. 19 Федерального конституционного закона от 31.12.1996 N 1-ФКЗ «О судебной системе Российской Федерации», Верховный суд РФ в целях обеспечения единообразного применения законодательства Российской Федерации дает судам разъяснения по вопросам судебной практики. Это выражается в подготовке обзоров судебной практики по разным отраслям, а иногда и конкретным статьям. Такие обзоры представляют собой краткое изложение дел и позиции суда по нему. Иногда в рамках одного вопроса рассматриваются сразу несколько однотипных споров, по которым делается обобщенный вывод.

Для совершенствования законодательства также необходимо изучать судебную практику. При этом можно заметить единообразные решения, которые вынуждены принимать суды из-за несовершенства законодательного акта.

Примером такой ситуации может служить отсутствие четкого закрепления сроков подачи искового заявления после отмены судебного приказа в делах о взыскании финансовых санкций с организации, данный пример включен в обзор судебной практики Верховного Суда Российской Федерации № 2 (2019)[[2]](#footnote-2). В этих ситуациях судам приходится принимать решения по аналогии, каждый раз указывая одни и те же причины, а также ссылаясь на позицию Верховного Суда РФ.

Подобные дела оборачиваются высокой экономической неэффективностью. Государство оплачивает расходы судов и других органов, сотрудники тратят большое количество времени на судебные процессы, которых могло и не быть. Хотя достаточно внести одну техническую норму в законодательный акт, чтобы прервать череду подобных процессов, снизить нагрузку на суды по аналогичным делам.

Чтобы использовать все вышеописанные преимущества судебной практики, необходимо ускорить процесс её изучения и обобщения. Для этого мы предлагаем использовать автоматизированную систему, архитектура которой рассматривается в данной статье. В результате использования системы вместо текста из нескольких страниц, юристы получат краткое содержание текста с наиболее значимой информацией, которую они могут использовать для принятия решения о необходимости подробного изучения документа или в каких-то других целях. Также юристам будет предоставлен граф связей между извлеченными концептами, ассоциативные связи между ними и статистическая информация.

**Гибридные интеллектуальные информационные системы**

В настоящее время для построения интеллектуальных систем используется большое количество подходов: продукционные правила, нейронные сети, нечеткая логика, эволюционные методы и др. При этом можно отметить явную тенденцию к совместному использованию разных методов для решения различных классов задач. Это привело к появлению такого направления, как «гибридные интеллектуальные системы» (ГИС). Основополагающими работами в области ГИС можно считать работы А.В. Колесникова [5, 6].

Ключевым вопросом является вопрос о том, каким образом реализовать принцип гибридности. Ответ на этот вопрос предлагается, в частности, в работе Н.Г. Ярушкиной [8]. В ней сформулирован следующий принцип гибридности [8, с. 20-21]: «В литературе встречаются схемы гибридизации нейроинформатики и ИИ, построенные по следующему принципу: правое полушарие – нейрокомпьютер; левое полушарие – основанная на знаниях система, а вопрос лишь в их взаимодействии или балансе право- и лево-полушарности. В реальном поведении человека невозможно разделить восприятие и логическую обработку, поэтому более успешной представляется схема глубинной интеграции».

В настоящее время интеллектуальные системы, как правило, не разрабатываются отдельно, но встраиваются в виде модулей в традиционные информационные системы для решения задач, связанных с интеллектуальной обработкой данных и знаний. Такую комбинированную систему в соответствии с [3] назовем гибридной интеллектуальной информационной системой (ГИИС). ГИИС обладает следующими особенностями:

* сочетает различные методы, используемые для построения интеллектуальных систем, и в этом смысле является ГИС;
* сочетает интеллектуальные методы с традиционными методами, используемыми для разработки данных в информационных системах, и в этом смысле является комбинацией ГИС и информационной системы, предназначенной для обработки данных.

Таким образом, под ГИИС будем понимать информационную систему, которая использует комбинацию традиционных методов обработки данных и интеллектуальных методов, что полностью соответствует концепции Н.Г. Ярушкиной [8]. Однако, на взгляд авторов статьи, вместо право- и лево-полушарности скорее стоит говорить о «подсознании» и «сознании» ГИИС. «Подсознание» строится на основе методов мягких вычислений, а «сознание» на основе традиционных методов обработки данных и знаний. На рис. 1 представлена обобщенная архитектура ГИИС, построенная на основе «сознания» и «подсознания».

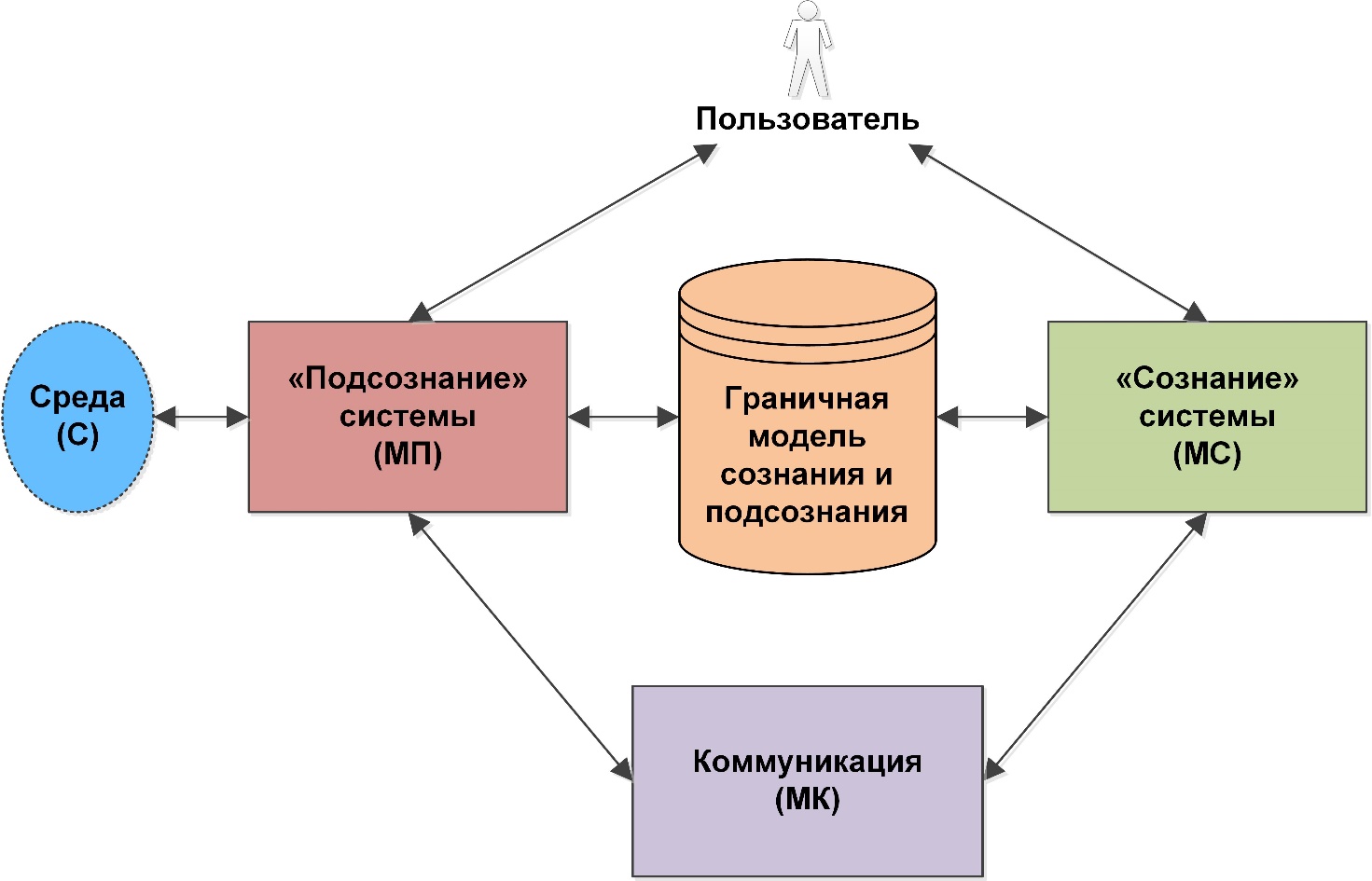


Рис. 1. Обобщенная архитектура ГИИС.

Основой системы являются «подсознание» системы (модуль подсознания, МП) и «сознание» системы (модуль сознания, МС). «Подсознание» связано со средой, в которой функционирует ГИИС.

Основной задачей МП является обеспечение взаимодействия ГИИС со «средой», или «выживание» ГИИС в среде.

Поскольку среда может быть представлена в виде набора непрерывных сигналов, то в качестве методов обработки данных «подсознания» хорошо подходят методы, основанные на нейронных сетях и нечеткой логике, в том числе и комбинированные нейронечеткие методы.

Модель данных «подсознания» максимально приближена к «понятийной системе» среды, представляет собой набор данных, который позволяет максимально эффективно взаимодействовать со средой. Часть этих данных может не иметь «физического смысла» с точки зрения МС, однако позволяет МП взаимодействовать со средой с нужной производительностью.

«Сознание» ГИИС строится на принципах обработки данных и знаний. Обработка данных в МС может вестись на основе традиционных языков программирования или технологии workflow. Однако, в последнее время все большую популярность приобретает подход на основе продукционных правил. Раньше данный подход использовался для принятия решения в экспертных системах, но в настоящее время на основе правил пишутся обычные программы. Такой подход называется программированием на основе правил (rule-based programming). К достоинствам подхода на основе правил можно отнести гибкость, так как в этом случае программа не кодируется жестко, а фактически «выводится» из правил на основе данных. К недостаткам можно отнести возможность зацикливания правил, а также сложность обработки большого объема правил.

Отметим, что задача хранения требуемых данных решается отдельно на уровне МС и на уровне МП. Мы предполагаем, что на уровне обобщенной архитектуры соответствующие хранилища «встроены» в МС и МП, поэтому хранилища не представлены явно на рис. 1.

Модуль сознания воспринимает понятийную систему как целостную модель «онтологического» класса и может «осознанно» обрабатывать элементы данной модели на основе правил.

Модуль подсознания воспринимает понятийную систему в виде отдельных (возможно несвязанных) признаков. Требования к «осознанию» целостности модели не предъявляется. Основным критерием является эффективность взаимодействия системы со средой.

С точки зрения коммуникации, в ГИИС возможны следующие варианты или их комбинации:

1. Коммуникация осуществляется через среду. МП читает данные из среды, преобразует и передает в МС. МС осуществляет логическую обработку и возвращает результаты обработки в МП. МП записывает результирующие данные в среду, откуда они могут быть прочитаны другими ГИИС.
2. Для коммуникации с другими ГИИС используется модуль коммуникации (МК). В зависимости от решаемых задач с МК может взаимодействовать МС (что характерно для традиционных информационных систем) или МП (что более характерно для систем на основе мягких вычислений).
3. Взаимодействие с пользователем также может осуществляться через МС (что характерно для традиционных информационных систем) или через МП (что может быть использовано, например, в автоматизированных тренажерах).

Граничная модель сознания и подсознания предназначена для глубинной интеграции модулей сознания и подсознания и представляет собой интерфейс между этими модулями с функцией хранения данных. В качестве данных выступает комплексная онтология, которая используется как сознанием, так и подсознанием. Основной задачей подсознания является распознавание из среды элементов онтологии. Если рассматривать сознание как разновидность экспертной системы, то распознанные элементы онтологии могут рассматриваться в качестве элементов операционной памяти экспертной системы, которые приводят к срабатыванию соответствующих правил. В зависимости от целей системы, правила могут формировать выходную информацию для пользователя или сигналы для модуля подсознания, которые оказывает требуемое воздействие на среду.

Предложенная архитектура рассматривается как обобщенный подход, который должен быть адаптирован для создания информационных систем в конкретных предметных областях.

**Использование подхода на основе сложных сетей для реализации ГИИС**

В современных информационных системах традиционно используются разнородные информационные модели данных, знаний и процессов. Эта ситуация сложилась исторически, потому что раньше мощность вычислительных систем была невысока и во главу угла ставилась производительность обработки информационной модели. Вопросы интеграции информационных систем и унификации информационных моделей оставались на втором плане.

В настоящее время ситуация изменилась. Появление и активное развитие технологий обработки больших данных, расширение круга информационно-аналитических задач привело к тому, что в качестве обрабатываемых данных вполне могут выступать знания, ситуации, процессы. Это требует новых подходов к интеграции систем, к информационным моделям.

Сервис-ориентированный подход (в том числе в его современном микросервисном варианте) до определенной степени решает задачу интеграции информационных систем, но при этом каждая система функционирует как черный ящик. Интеграция возможна только на уровне элементов, которые вынесены в интерфейс сервиса. Например, если в целях решения информационно-аналитических задач, мы хотим использовать OLAP-куб для хранения и агрегации не чисел, а ситуаций или процессов, то использование сервис-ориентированного подхода не поможет, требуется интеграция на уровне единой информационной модели.

В качестве интеграционной модели для реализации ГИИС мы предлагаем использовать метаграфовую модель.

Основополагающими работами по теории метаграфов являются работы А. Базу и Р. Блэннинга, которые в 2007 году были обобщены в виде монографии [1]. На кафедре «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана в рамках в рамках развития данного направления метаграфовую модель предлагается применять как средство для описания сложных сетей [2], как средство для описания семантики и прагматики информационных систем [9], как средство для описания гибридных интеллектуальных информационных систем [3].

Для обработки метаграфовой модели предлагается использовать подход на основе многоагентной системы (МАС). В соответствии с [12], под программным агентом будем понимать программный модуль, который выполняется в виде автономной задачи (не зависит от других агентов), способен обмениваться информацией со средой и другими агентами. Под МАС будем понимать систему однородных или разнородных агентов, функционирующих в среде.

Для реализации ГИИС наиболее интересным представляется подход на основе холонической многоагентной системы (холонической МАС). В соответствии с определением [12, с. 234] холон – это «целое, рассматриваемое в то же время как часть целого». С точки зрения данного подхода, рассмотренные компоненты, такие как МП, МС, МК являются агентами. В то же время они являются частями системы, которая, в свою очередь, является агентом.

При этом МП является сложной структурой, которая включает агенты нижнего уровня, каждый из которых может в свою очередь включать МП, МС, МК, предназначенные для решения конкретных задач данного агента. Несмотря на то, что агент нижнего уровня находится в составе МП, он может включать в свою структуру МС, предназначенный для решения задач МП более высокого уровня. Поэтому, с точки зрения данного подхода, нет ничего удивительного в том, что в МС могут использоваться нечеткие продукционные правила, а в МП входят «классические» модули обработки данных.

**Метаграфовая модель описания данных, знаний, и процессов ГИИС**

Под метаграфом будем понимать следующую структуру:  где MG – метаграф; *V* – множество вершин метаграфа; *MV* – множество метавершин метаграфа; *E* – множество ребер метаграфа; *ME* – множество метаребер метаграфа.

Вершина метаграфа характеризуется множеством атрибутов:  где *vi* – вершина метаграфа; *atrk* – атрибут.

Ребро метаграфа характеризуется множеством атрибутов, исходной и конечной вершиной и признаком направленности:  где *ei* – ребро метаграфа; *vS* – исходная вершина (метавершина) ребра; *vE* – конечная вершина (метавершина) ребра; *eo* – признак направленности ребра (*eo=true* – направленное ребро, *eo=false* – ненаправленное ребро); *atrk* – атрибут.

Фрагмент метаграфа:  где *MGi* – фрагмент метаграфа; *evj* – элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа.

Таким образом, фрагмент метаграфа в общем виде может содержать произвольные вершины (метавершины) и ребра (метаребра) без ограничений. Ограничения вводятся на фрагменты метаграфа, входящие в метавершину и метаребро.

Метавершина метаграфа является основным элементом предлагаемой модели:  где *mvi* – вершина метаграфа; *atrk* – атрибут, *evj* – элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа.

Таким образом, метавершина в дополнение к свойствам вершины включает вложенный фрагмент метаграфа. При этом ребра и метаребра этого фрагмента могут быть только ненаправленными, *eo=false*.

Метаребро метаграфа:   где *mei* – метаребро метаграфа; *vS* – исходная вершина (метавершина) ребра; *vE* – конечная вершина (метавершина) ребра; *eo* – признак направленности метаребра (*eo=true* – направленное метаребро, *eo=false* – ненаправленное метаребро); *atrk* – атрибут; *evj* – элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа.

Таким образом, метаребро в дополнение к свойствам ребра включает вложенный фрагмент метаграфа. При этом ребра и метаребра этого фрагмента могут быть только направленными, *eo=true*.

Определения метавершины и метаребра являются рекурсивными, так как элементы *evj* могут быть, в свою очередь, метавершинами и метаребрами. Метавершина является формализмом описания данных, а метаребро – формализмом описания процессов.

Наличие у метавершин собственных атрибутов и связей с другими вершинами является важной особенностью метаграфов. Это соответствует принципу эмерджентности, то есть приданию понятию нового качества, несводимости понятия к сумме его составных частей. Фактически, как только вводится новое понятие в виде метавершины, оно «получает право» на собственные свойства, связи и т.д., так как в соответствии с принципом эмерджентности новое понятие обладает новым качеством и не может быть сведено к подграфу базовых понятий.

Таким образом, метаграф можно охарактеризовать как «сеть с эмерджентностью», то есть фрагмент сети, состоящий из вершин и связей, который может выступать как отдельное целое.

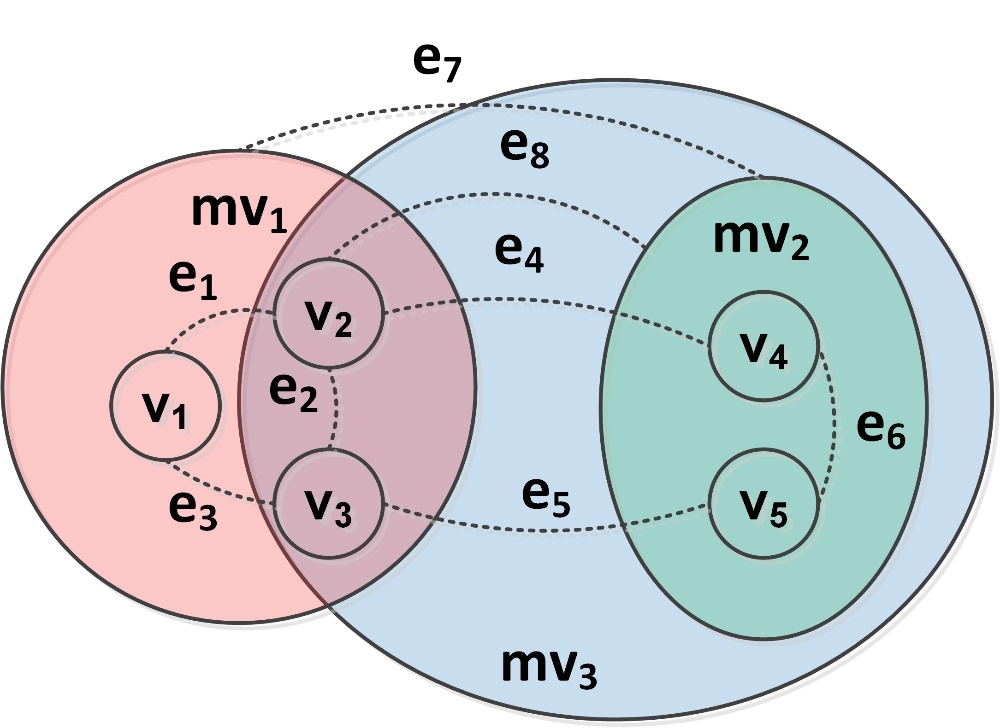


Рис. 2. Пример описания метавершины метаграфа.

Пример описания метавершины метаграфа показан на рис. 2. Данный метаграф содержит вершины, метавершины и ребра. На рис. 2 показаны три метавершины: *mv1*, *mv2* и *mv3*. Метавершина *mv1* включает вершины *v1*, *v2*, *v3* и связывающие их ребра *e1*, *e2*, *e3*. Метавершина *mv2* включает вершины *v4*, *v5* и связывающее их ребро *e6*. Ребра *e4*, *e5* являются примерами ребер, соединяющих вершины *v2-v4* и *v3-v5*, включенные в различные метавершины *mv1* и *mv2*. Ребро *e7* является примером ребра, соединяющего метавершины *mv1* и *mv2*. Ребро *e8* является примером ребра, соединяющего вершину *v2* и метавершину *mv2*. Метавершина *mv3* включает метавершину *mv2*, вершины *v2*, *v3* и ребро *e2* из метавершины *mv1* а также ребра *e4*, *e5*, *e8*, что показывает холоническую структуру метаграфа.

Если метавершины предназначены прежде всего для описания данных и знаний, то метаребра предназначены в большей степени для описания процессов.

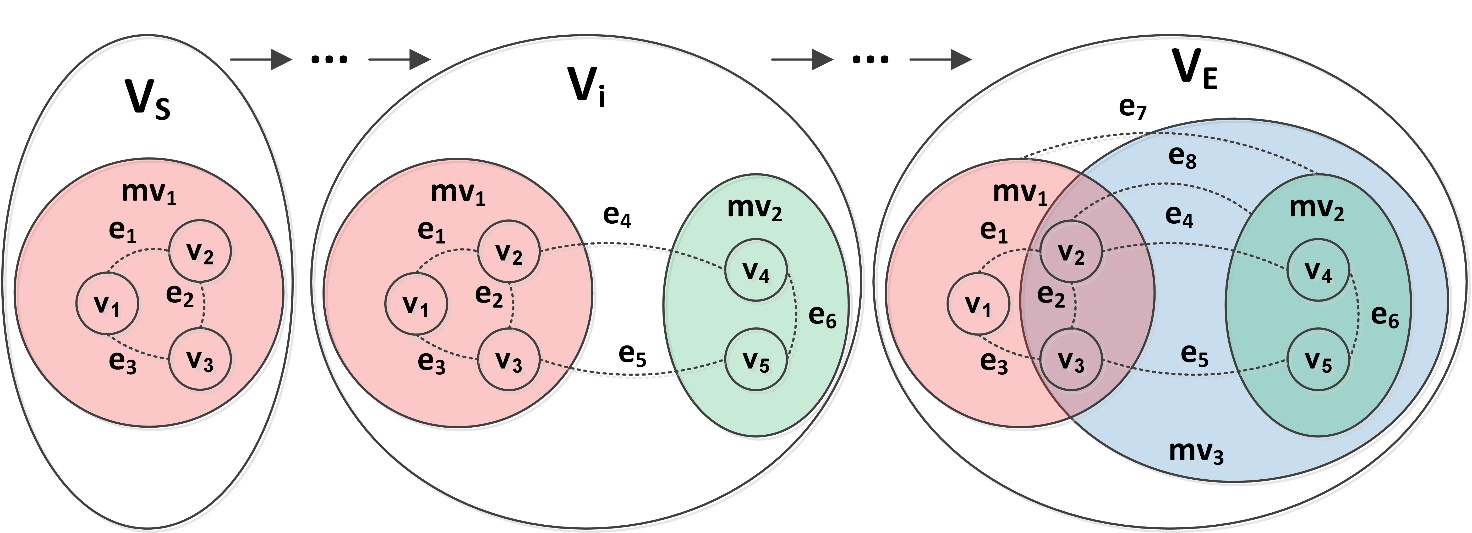


Рис. 3. Пример описания метаребра метаграфа.

Пример описания метаребра метаграфа представлен на рис. 3. Метаребро содержит метавершины *vS*, … *vi*, … *vE* и связывающие их ребра. Исходная метавершина содержит фрагмент метаграфа. В процессе преобразования исходной метавершины *vS* в конечную метавершину *vE* происходит дополнение содержимого метавершины, добавляются новые вершины, связи, вложенные метавершины.

Таким образом, метаграфовая модель позволяет в рамках единой модели описывать данные, знания и процессы.

**Обработка метаграфовой модели с использованием метаграфовых агентов**

Для обработки метаграфовой модели используются два вида агентов: агент-функция и метаграфовый агент.

Определим агент-функцию следующим образом:  где *agF* – агент-функция; *MGIN* – метаграф, который выполняет роль входного параметра агента-функции; *MGOUT* – метаграф, который выполняет роль выходного параметра агента-функции; *AST* – абстрактное синтаксическое дерево агента-функции, которое может быть представлено в виде метаграфа.

Определим метаграфовый агент следующим образом:  где *agM* – метаграфовый агент; *MGD* – метаграф данных и знаний, на основе которого выполняются правила агента; *R* – набор правил (множество правил *rj*); *AGST* – стартовое условие выполнения агента (фрагмент метаграфа, который используется для стартовой проверки правил, или стартовое правило).

Структура правила метаграфового агента:  где *ri* – правило; *MGj* – фрагмент метаграфа, на основе которого выполняется правило; *OPMG* – множество операций, выполняемых над метаграфом.

Антецедентом правила является фрагмент метаграфа, консеквентом правила является множество операций, выполняемых над метаграфом.

Пример представления метаграфового агента в виде фрагмента метаграфа приведен на рис. 4.

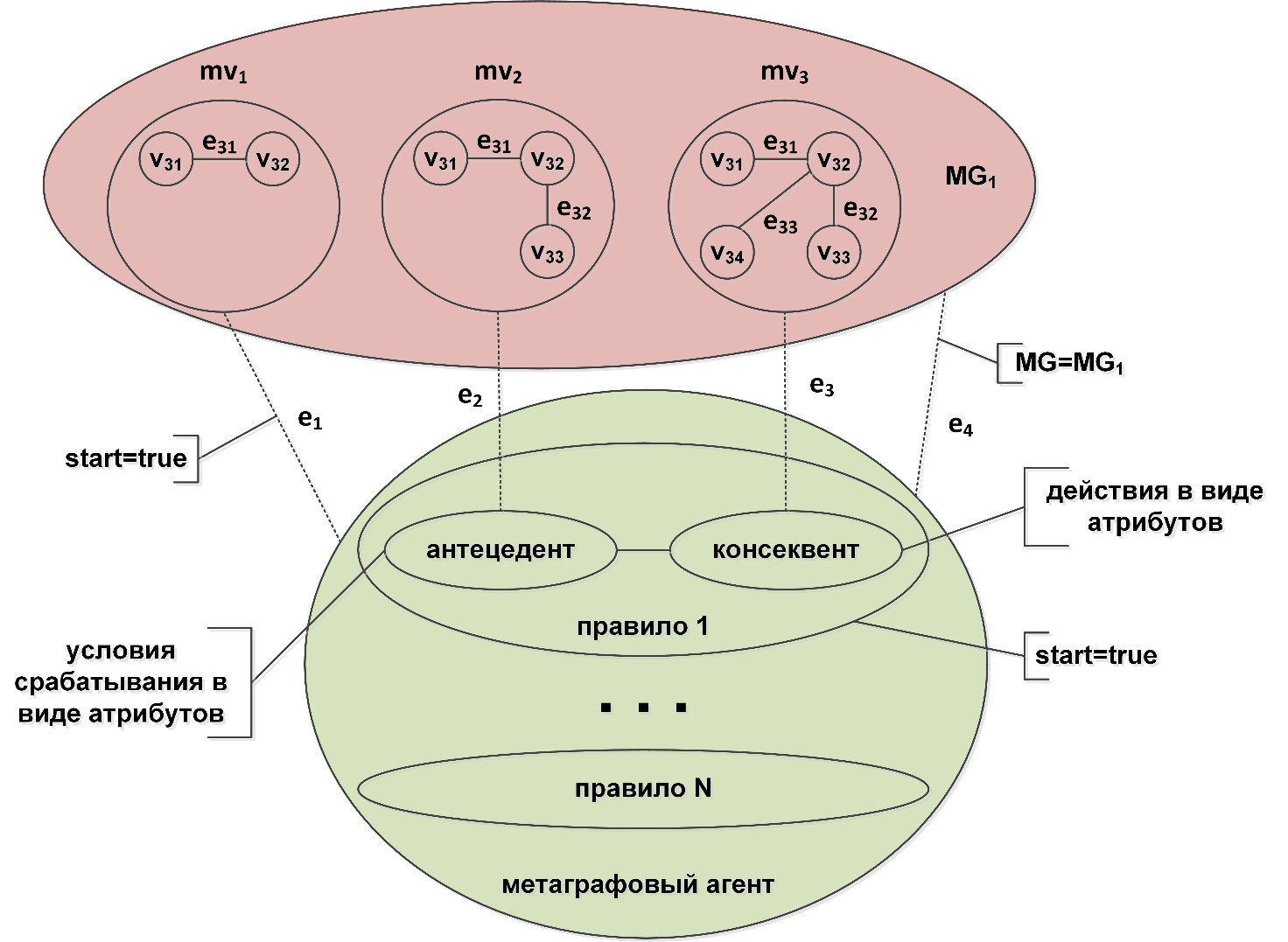


Рис. 4. Представление метаграфового агента в виде фрагмента метаграфа.

Метаграфовый агент представлен в виде метавершины метаграфа. В соответствии с определением, он связан с метаграфом *MG1*, на основе которого выполняются правила агента. Данная связь показана с помощью ребра *e4*.

Метаграфовый агент содержит множество вложенных метавершин, соответствующих правилам (правило 1 – правило N). Каждая метавершина правила содержит вершины антецедента и консеквента. В данном примере с антецедентом правила связана метавершина данных *mv2*, что показано ребром *e2*, а с консеквентом правила связана метавершина данных *mv3*, что показано ребром *e3*. Условия срабатывания антецедента и множество действий консеквента задаются в виде атрибутов соответствующих вершин.

Стартовое условие выполнения агента задается с помощью атрибута «start=true». Если стартовое условие задается в виде стартового правила, то данным атрибутом помечается метавершина соответствующего правила, в данном примере это правило 1. Если стартовое условие задается в виде стартового фрагмента метаграфа, который используется для стартовой проверки правил, то атрибутом «start=true» помечается ребро, которое связывает стартовый фрагмент метаграфа с метавершиной агента, в данном примере это ребро *e1*.

Под «активным узлом метаграфа»  будем понимать вершину или метавершину метаграфа, с которой связаны программные агенты, выполняющие определенные функции. Под «метаграфовым процессом» будем понимать метаребро, базирующееся на активных узлах:  где *PROCi* – метаграфовый процесс; *vS* – исходная вершина (метавершина) процесса; *vE* – конечная вершина (метавершина) процесса; *atrk* – атрибут; *MGj* – фрагмент метаграфа, такой, что каждая его вершина или метавершина *mvnode* является активным узлом метаграфа. На рис. 5 представлен пример метаграфового процесса.

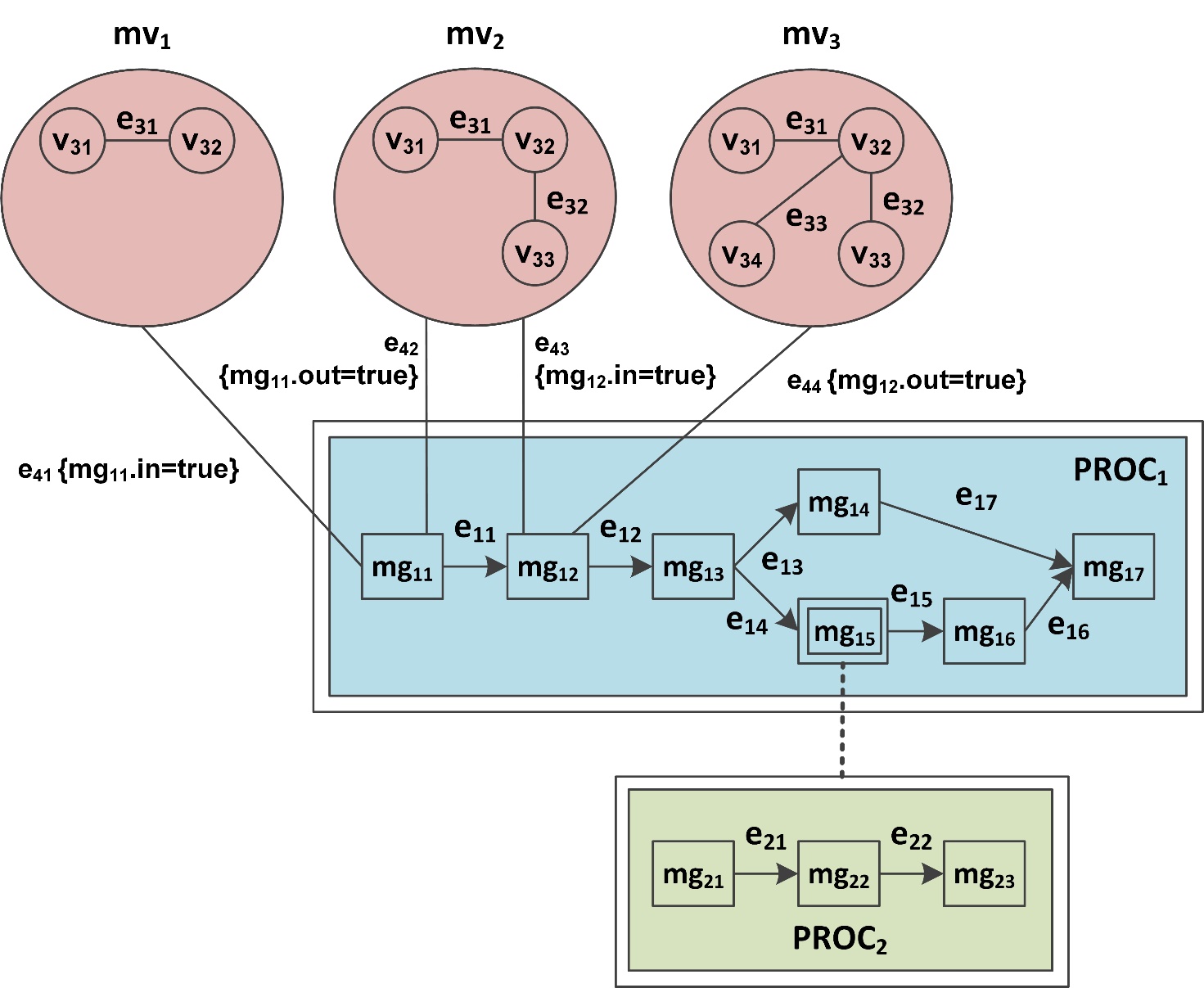


Рис. 5. Пример метаграфового процесса.

На рис. 5 окружностями показаны вершины и метавершины, используемые для описания данных. Прямоугольниками показаны метавершины или метаребра, соответствующие элементам процесса. Метаграфовые процессы *PROC1* и *PROC2* показаны двойными прямоугольниками. Ненаправленными связями показаны ненаправленные ребра (eo=false), а направленными стрелками показаны направленные ребра (eo=true). Пунктирной связью показана вложенность фрагмента метаграфа.

Процесс состоит из элементов процесса *v1\** (под «\*» понимается произвольное значение второго индекса). Элементы процесса соединены направленными ребрами *e1\**. В качестве примера для элемента процесса *v15* показаны элементы вложенного подпроцесса. Таким образом, элемент *v15* одновременно является и элементом процесса me1 и метаребром, которое содержит вложенный процесс.

Данные, поступающие на вход элемента *v11*, показаны в виде метавершины mv1, которая содержит вложенные вершины данных *v31* и *v32* и ненаправленную связь между ними *e31*. Связь метавершины *mv1* с элементом процесса *v11* осуществляется с помощью ребра *e41*. Признаком того, что метавершина *mv1* содержит входные данные процесса *v11*, моделируется с помощью атрибута ребра *e41* (вершины и ребра могут иметь атрибуты, так как используется модель атрибутивного метаграфа). В данном случае используется атрибут *v11.in=true*. Аналогично с использованием ребер *e42*, *e43*, *e44* производится привязка метавершин *mv2* и *mv3* к элементам процесса *v11* и *v12* в качестве входных-выходных данных.

На рис. 5 показан случай, когда выходные данные предыдущего процесса являются входными данными следующего процесса. Однако, предлагаемая модель связи метавершин данных с элементами процесса носит более гибкий характер и позволяет моделировать передачу данных как через входные-выходные метавершины данных, так и другими способами, например, через представленный метавершиной общий контекст.

**Предлагаемая архитектура системы**

Рассмотрим предлагаемую архитектуру информационной системы анализа судебной практики арбитражных судов, построенную на основе подхода ГИИС.

Классическая архитектура практически любой аналитической информационной системы включает подсистему сбора данных и подсистему обработки данных. В классической архитектуре при проектировании подсистемы сбора, в основном, рассматриваются аспекты сбора данных из различных источников, меньшее значение уделяется унификации семантики данных. Подсистема обработки формирует аналитику на основе собранных данных.

Напротив, в архитектуре ГИИС основной упор делается на семантику данных. В концепции ГИИС подсистеме сбора можно сопоставить подсознание, а подсистеме обработки сопоставить сознание, но такое сопоставление достаточно условно. Основная задача подсознания – формирование концептов, используемых для принятия решения. Основная задача сознания – принятие решения на основе выделенных концептов. Архитектура предлагаемой системы представлена на рис. 6.

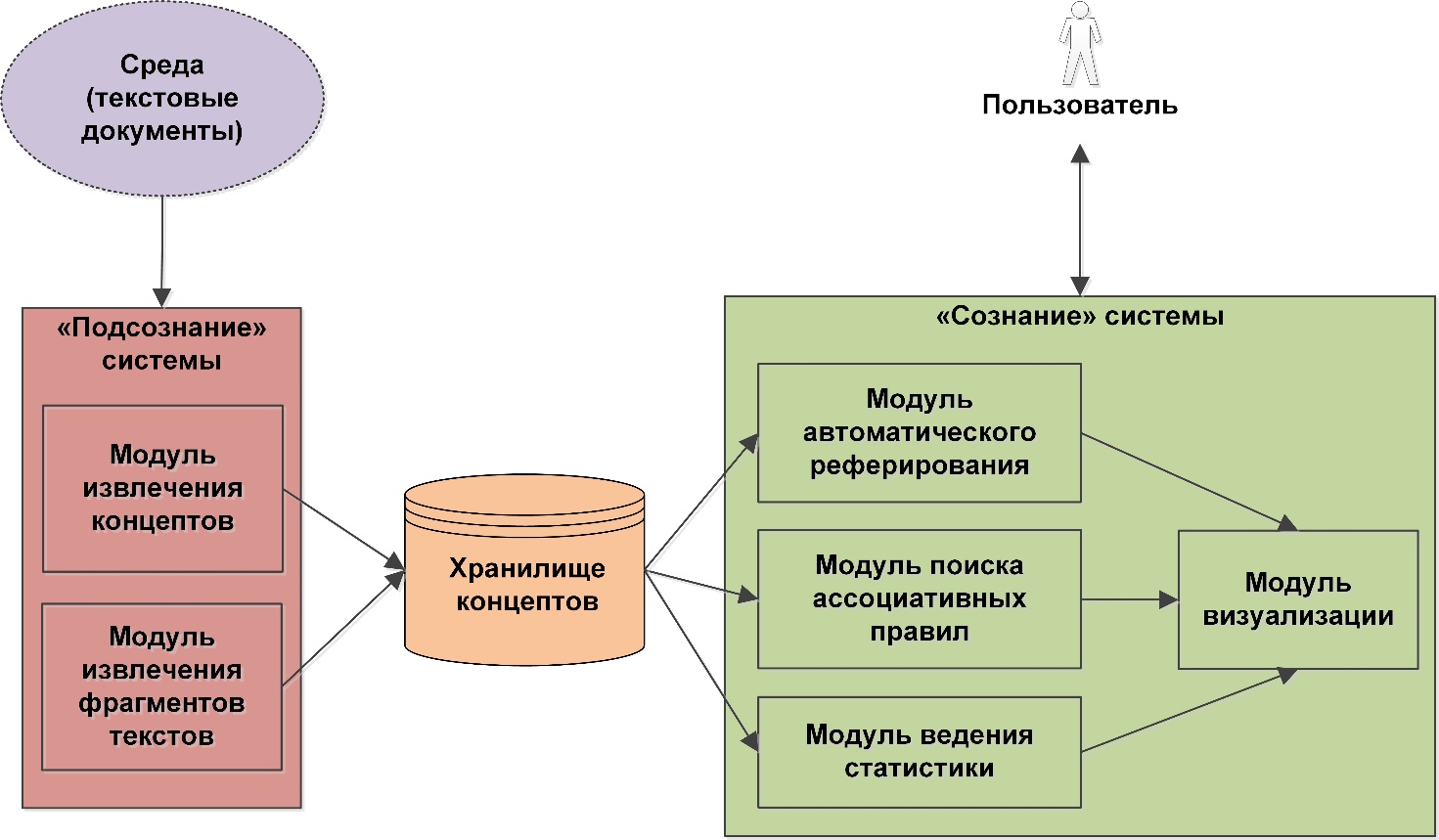


Рис. 6. Архитектура предлагаемой системы.

Средой в данной системе являются исходные текстовые документы юридической направленности.

Подсознание системы включает модуль извлечения концептов и модуль извлечения фрагментов текстов, при этом фрагменты текстов могут рассматриваться как укрупненные концепты, используемые при формировании аналитики.

Извлеченные концепты и фрагменты текстов помещаются в хранилище концептов, которое соответствует граничной модели сознания и подсознания.

Сознание системы включает модули автоматического реферирования, визуализации, ведения статистики, поиска ассоциативных правил.

**Модули подсознания системы**

В модуле извлечения концептов из текста извлекаются основные данные:

* имя или название истца и ответчика;
* имя или название заявителя (в жалобах);
* суть дела;
* цена иска;
* решение суда;
* и другие.

Модуль строится на гибридном подходе, сочетая в себе как подход на основе правил, так и использование машинного обучения. Отметим, что такой подход хорошо зарекомендовал себя в обработке текстов [10]. Для тривиальных случаев используются правила. Например, тип документа, цена иска, решение суда и т.д. Остальной текст анализируются нейросетевыми моделями на основе фреймворка TensorFlow [4].

В зависимости от извлекаемых данных, требуется решить различные задачи машинного обучения, а именно:

* извлечение именованных сущностей;
* извлечение фактов;
* извлечение временных данных.

Для каждой задачи создается свой подмодуль с нейросетью и/или правилами. Для обучения моделей можно использовать датасеты не только на основе судебной практики, так как извлечение именованных сущностей и временных данных носят универсальный характер. Однако, специальный размеченный датасет со специализированными текстами может дать более высокое качество.

Модуль извлечения фрагментов текстов отвечает за отбор текстовых блоков, которые могут быть использованы в реферате. Для решения этой задачи используются методы машинного обучения [7].

С точки зрения машинного обучения предсказание подходящего текстового блока можно отнести к задаче обучения с учителем (по прецедентам), а именно многоклассовой классификации. У каждого объекта есть определенный набор признаков. В нашем случае это абзац и некоторые дополнительные данные, которые извлекаются на предыдущем шаге. Ответом является метка класса, к которому отнесен блок.

После классификации остаются только блоки, отнесенные к полезным. В хранилище концептов передается уже частично сжатый текст.

**Модули сознания системы**

Модуль автоматического реферирования является основным инструментом, который значительно ускоряет для юриста обработку документов. В [11] отмечается, что методы автоматического реферирования можно разделить на две большие группы:

* Экстракция (извлечение предложений, Sentence Extraction, квазиреферирование) – извлечение из исходного текста наиболее важных и существенных информационных блоков (абзацев, предложений).
* Абстракция (извлечение содержания, Content Extraction) – генерация реферата с порождением нового текста, содержательно обобщающего первичный документ или документы.

Для методов, основанных на абстракции, в большей степени подходят концепты, извлеченные с помощью модуля извлечения концептов. Для методов, основанных на экстракции, могут быть использованы фрагменты текстов, полученные на выходе модуля извлечения фрагментов текстов.

Следует отметить, что методы, основанные на абстракции, вызывают все больший интерес, а для решения задачи реферирования используются методы на основе графов.

Сформированный граф документа может быть представлен юристу с использованием модуля визуализации.

Сокращение объема текста далеко не единственная задача, которую должна решать информационная система. Не менее важной задачей является выявление причин принятия судебных решений. Здесь на помощь приходит модуль поиска ассоциативных правил, который позволяет понять какие факторы привели к принятию конкретного судебного решения. Необходимо отметить, что практически все существующие алгоритмы поиска ассоциативных правил предполагают работу с концептами, поэтому очень важно то, что подсознание ГИИС выделяет не просто текст, а концепты, которые поступают на вход данного модуля.

Ассоциативные правила также могут быть представлены в виде графов с использованием модуля визуализации.

Модуль ведения статистики позволяет хранить и обрабатывать количественную информацию: сколько дел было обработано в том или ином суде за определенный временной период и т.д. Для реализации этого модуля может быть использована OLAP-система, в этом случае она может иметь собственной хранилище. Хранилище концептов может быть использовано для формирования и иерархической организации измерений OLAP-системы.

Статистические данные в графическом представлении также отображаются с использованием модуля визуализации.

**Выводы**

Рассмотрены особенности автоматизации документов судебной практики. Предложена возможная архитектура автоматизированной системы реферирования судебной практики арбитражных судов.

Обобщенную структуру ГИИС предлагается строить на основе модулей «сознания» и «подсознания». На основе обобщенной структуры могут быть построены частные случаи структуры ГИИС, которым соответствуют конкретные ГИИС.

Для реализации ГИИС предполагается использовать холоническую МАС. Структура такой МАС может быть описана с использованием метаграфового подхода.

Метаграфовый подход является одним из вариантов описания «сетей с эмерджентностью». Эмерджентность обеспечивается за счет использования метавершин.

Для обработки метаграфовой модели используются два вида агентов: агент-функция и метаграфовый агент.

C активным узлом метаграфа связаны программные агенты, выполняющие определенные функции обработки метаграфовой модели. Под «метаграфовым процессом» будем понимать метаребро, базирующееся на активных узлах:

Предлагаемая архитектура информационной системы анализа судебной практики арбитражных судов базируется на концепции ГИИС. Средой в данной системе являются исходные текстовые документы юридической направленности.

Подсознание системы включает модуль извлечения концептов и модуль извлечения фрагментов текстов, при этом фрагменты текстов могут рассматриваться как укрупненные концепты, используемые при формировании аналитики.

Извлеченные концепты и фрагменты текстов помещаются в хранилище концептов, которое соответствует граничной модели сознания и подсознания.

Сознание системы включает модули автоматического реферирования, визуализации, ведения статистики, поиска ассоциативных правил.

С использованием метаграфовой модели возможно унифицированное представление архитектуры разрабатываемой системы.

**Литература**

1. Basu A., Blanning R. Metagraphs and their applications. Springer, 2007. 174 p.

2. Черненький В.М., Терехов В.И., Гапанюк Ю.Е. Представление сложных сетей на основе метаграфов // Нейроинформатика-2016. XVIII Всероссийская научно-техническая конференция. Сборник научных трудов. Ч. 1. М.: НИЯУ МИФИ, 2016. C. 173-178.

3. Черненький В.М., Терехов В.И., Гапанюк Ю.Е. Структура гибридной интеллектуальной информационной системы на основе метаграфов. Нейрокомпьютеры: разработка, применение. 2016. Выпуск №9. С. 3-14.

4. Geron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. O’Reilly, 2019. 856 p.

5. Колесников А.В. Гибридные интеллектуальные системы. Теория и технология разработки. – СПб.: СПбГТУ, 2001. – 137 с.

6. Колесников А.В., Кириков И.А., Листопад С.В. Гибридные интеллектуальные системы с самоорганизацией: координация, согласованность, спор. – М.: ИПИ РАН, 2014. – 189 с.

7. Kulkarni A., Shivananda A. Natural Language Processing Recipes. Unlocking Text Data with Machine Learning and Deep Learning using Python. Apress, 2019. 234 p.

8. Прикладные интеллектуальные системы, основанные на мягких вычислениях. / под ред. Н.Г. Ярушкиной. – Ульяновск: УлГТУ, 2004. – 139 с.

9. Самохвалов Э.Н., Ревунков Г.И., Гапанюк Ю.Е. Использование метаграфов для описания семантики и прагматики информационных систем. Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». 2015. Выпуск №1. С. 83-99.

10. Taran M., Revunkov G., Gapanyuk Yu. The Hybrid Intelligent Information System for Poems Generation. Neuroinformatics 2019: Advances in Neural Computation, Machine Learning, and Cognitive Research III. pp. 78-86.

11. Тарасов С.Д. Современные методы автоматического реферирования. Научно-технические ведомости СПбГПУ, 2010. Выпуск №6. С. 59-74.

12. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.

**The architecture of the hybrid intelligent information system for analysis of judicial practice of arbitration courts**

Maria Taran, Ph.D. student at the Bauman Moscow State Technical University, Moscow, Russian Federation.

Yuriy Gapanyuk, Ph.D., Associate Professor of the Bauman Moscow State Technical University, Moscow, Russian Federation.

E-mail: gapyu@bmstu.ru

**Keywords:** arbitration court, Hybrid Intelligent System (HIS), Hybrid Intelligent Information System (HIIS), the conscious of information system, the subconscious of information system, text mining, multi-agent system (MAS), graph, metagraph, metavertex, metaedge, metagraph process, metagraph agent, active metagraph node.

**Abstract**

**Purpose of the work:** improvement of the scientific and methodological base of information support of the system of judicial practice of arbitration courts.

**Methods used:** methods of designing hybrid intelligent information systems.

**Results obtained:** the approach to the development of Hybrid Intelligent Information Systems (HIIS) based on the use of modules of consciousness, subconsciousness, and communication is considered. An approach based on complex networks for the implementation of HIIS is considered; it is shown that the metagraph model allows describing data, knowledge, and processes as components of the HIIS. The structure of the metagraph agent providing processing of the metagraph data model is considered. It is shown that a metagraph agent can be represented by a metagraph model, which allows top-level metagraph agents to modify the structure of lower-level metagraph agents. The concept of an active metagraph node is introduced as a combination of a metagraph data model and a metagraph agent model. Based on the active metagraph node, the concept of a metagraph process is introduced. As a result, the architecture of a hybrid intelligent information system for the analysis of the judicial practice of arbitration courts is proposed.

**References**

1. Basu A., Blanning R. Metagraphs and their applications. Springer, 2007. 174 p.

2. Chernenkiy V.M., Terekhov V.I., Gapanyuk Yu.E. Predstavleniye slozhnikh setey na osnove metagrafov [Metagraph representation of complex networks]. Trudi XVIII vserossiyskoy konferencii “Neuroinformatics-2016” [Proc. XVIII all-russian conference “Neuroinformatics-2016”], Moscow, 2016, pp. 173-178.

3. Chernenkiy V.M., Terekhov V.I., Gapanyuk Yu.E. Struktura gibridnoy intellektualnoy informacionnoy sistemy na osnove metagrafov [Structure of the hybrid intelligent information system based on metagraphs]. Neirokompyutery: razrabotka, primeneiye [Neurocomputers: development, application], 2016, no. 9, pp. 3-14.

4. Geron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. O’Reilly, 2019. 856 p.

5. Kolesnikov A.V. Gibridniye intellectualniye sistemy. Teoriya i tehnologiya razrabotki [Hybrid intelligent systems. Development theory and technology]. Saint Petersburg, SPbGTU Publ., 2001, 137 p.

6. Kolesnikov A.V., Kirikov I.A., Listopad S.V. Gibridniye intellectualniye sistemy s samoorganizatsiey: koordinatsia, soglasovannost, spor [Hybrid intelligent systems with self-organization: coordination, coherence, argument]. Moscow, IPI RAN Publ., 2014, 189 p.

7. Kulkarni A., Shivananda A. Natural Language Processing Recipes. Unlocking Text Data with Machine Learning and Deep Learning using Python. Apress, 2019. 234 p.

8. Prikladnie intellektualnie sistemi, osnovannie na myagkih vichisleniyakh [Applied intelligent systems based on soft computing] Ed. by N.G. Yarushkina. Ulianovsk, UlGTU Publ., 2004, 139 p.

9. Samokhvalov E.N., Revunkov G.I. Gapanyuk Yu.E. Ispolzovaniye metagraphov dlya opisaniya semantiki i pragmatiki informatsionnykh sistem [Metagraphs for information systems semantics and pragmatics definition]. Vestnik MGTU im. N.E. Baumana, seriya “Priborostroeniye” [Herald of the Bauman Moscow State Technical University, “Instrument Engineering”], 2015, no. 1, pp. 83-99.

10. Taran M., Revunkov G., Gapanyuk Yu. The Hybrid Intelligent Information System for Poems Generation. Neuroinformatics 2019: Advances in Neural Computation, Machine Learning, and Cognitive Research III. pp. 78-86.

11. Tarassov S.D. Sovremennyie metody avtomaticheskogo referirovaniya [Modern methods of automatic summarization]. Nauchno-tehnicheskiye vedomosti SPbGTU, 2010. Issue №6. P. 59-74.

12. Tarassov V.B. Ot mnogoagentnykh sistem k intellektual’nym organizatsiyam: filosofiya, psikhologiya, informatika [From multi-agent systems to intelligent organizations: philosophy, psychology, computer science]. Moscow, Editorial URSS Publ., 2002. 352 p.

1. Сайт «Электронное правосудие». – URL: http://ras.arbitr.ru [↑](#footnote-ref-1)
2. Обзор судебной практики Верховного Суда Российской Федерации. – URL: http://www.supcourt.ru/documents/practice/?year=2019 [↑](#footnote-ref-2)