

АРХИТЕКТУРА ГИБРИДНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ АНАЛИЗА СУДЕБНОЙ АРБИТРАЖНОЙ ПРАКТИКИ

Таран М.О., Гапанюк Ю.Е.*

Ключевые слова: арбитражный суд, гибридная интеллектуальная система, гибридная интеллектуальная информационная систем, сознание информационной системы, подсознание информационной системы, анализ текстов, многоагентная система, 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; исследован подход на основе сложных сетей для реализации ГИИС; показано, что метаграфовая модель позволяет описывать данные, знания и процессы, как составные части ГИИС; исследована структура метаграфового агента, обеспечивающего обработку метаграфовой модели; показано, что метаграфовый агент можно представить 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; обоснована архитектура ГИИС анализа судебной практики арбитражных судов.

DOI: 10.21681/1994-1404-2020-1-15-25

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 завершение 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 играют ключевую роль при подготовке к судебному процессу. Из-за законодательных коллизий или отсутствия регулирующих норм большинство судебных решений основывается на правовых позициях Пленумов Верховного Суда РФ (и/или Высшего Арбитражного Суда РФ до упразднения в 2014 г.).

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.

* **Таран Мария Олеговна**, аспирант Московского государственного технического университета им. Н. Э. Баумана, Российская Федерация, г. Москва.

E-mail: gapyu@bmstu.ru

Гапанюк Юрий Евгеньевич, кандидат технических наук, доцент Московского государственного технического университета им. Н. Э. Баумана, Российская Федерация, г. Москва.

E-mail: gapyu@bmstu.ru

По информации сайта «Электронное правосудие», за девять месяцев 2019 г. в арбитражных судах России было выпущено более 1 300 000 документов¹. Обработать, обобщить и сделать какие-либо выводы из такого массива документов в ручном режиме достаточно сложно.

В соответствии с ч. 4 ст. 19 Федерального конституционного закона от 31 декабря 1996 г. № 1-ФКЗ «О судебной системе Российской Федерации», Верховный суд РФ в целях обеспечения единообразного применения законодательства РФ 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 № 2 (2019)² судебной практики Верховного Суда Российской Федерации. 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 было бы внести соответствующие технико-правовые нормы в законодательство, чтобы прервать череду подобных процессов, снизить нагрузку на суды по аналогичным делам [3].

Для продуктивного использования судебной практики представляется целесообразным ускорить процесс её изучения и обобщения [4]. Это возможно, в частности, на основе применения разработанной *автоматизированной системы*, архитектура которой рассматривается в данной статье. В результате использования системы вместо текста в несколько страниц юристы могут получить краткое содержание текста с наиболее значимой информацией, которую можно использовать для принятия решения о необходимости подробного изучения документа или в каких-то других целях. Юристам будет также 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) [1, 2].

Ключевым вопросом является вопрос о том, каким образом реализовать принцип гибридности. Ответ на этот вопрос предлагается, в частности, в работе [6]. В ней сформулирован следующий принцип гибридности [6, с. 20—21]: «В литературе встречаются схемы гибридации нейроинформатики и искусственного интеллекта, 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 [12] назовем гибридной интеллектуальной информационной системой (ГИИС), которая 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 [6]. Однако, по мнению 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».

¹ Сайт «Электронное правосудие». URL: <http://ras.arbitr.ru>

² Обзор судебной практики Верховного Суда Российской Федерации. URL: <http://www.supcourt.ru/documents/ practice/?year=2019>.



Рис. 1. Обобщенная архитектура ГИИС

Основой системы являются «подсознание» (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 известной технологии «потока работ» (англ. *“workflow”*). Однако в последнее время все большую популярность приобретает подход на основе продукционных правил. Раньше данный подход использовался для принятия решения в экспертных системах [5], но в настоящее время на основе правил пишутся обычные программы. Такой подход называется программированием на основе правил (англ. *“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 воз-

можность закликивания правил, а также сложность обработки большого объема правил.

Заметим, 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 на рис. 1 хранилища явно не представлены.

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. Требование «осознания» целостности модели не предъявляется. Основным критерием является эффективность взаимодействия системы со средой.

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

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.

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).

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 основа обобщенного подхода, 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* [10], 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; потребуется интеграция на уровне единой информационной модели.

В качестве интеграционной модели для реализации ГИИС мы предлагаем использовать *метаграфовую модель* [13].

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 [11], семантики и прагматики информационных систем [7], гибридных интеллектуальных информационных систем [12].

Для обработки метаграфовой модели предлагается использовать подход на основе многоагентной системы (МАС). В соответствии с [9], 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 [9, 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 в ГИИС

Под метаграфом будем понимать следующую структуру:

$$MG = \langle V, MV, E, ME \rangle,$$

где MG — метаграф; V — множество вершин метаграфа; MV — множество метавершин метаграфа; E — множество ребер метаграфа; ME — множество метаребер метаграфа.

Вершина метаграфа характеризуется множеством атрибутов:

$$v_i = \{atr_k\}, v_i \in V,$$

где v_i — вершина метаграфа; atr_k — атрибут.

Ребро метаграфа характеризуется множеством атрибутов, исходной и конечной вершиной и признаком направленности:

$$e_i = \langle v_s, v_e, eo, \{atr_k\} \rangle, e_i \in E, eo = true \mid false,$$

где e_i — ребро метаграфа; v_s — исходная вершина (метавершина) ребра; v_e — конечная вершина (метавершина) ребра; eo — признак направленности ребра ($eo=true$ — направленное ребро, $eo=false$ — ненаправленное ребро); atr_k — атрибут.

Фрагмент метаграфа:

$$MG_i = \{ev_j\}, ev_j \in (V \cup E \cup MV \cup ME),$$

где MG_i — фрагмент метаграфа; ev_j — элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа.

Отсюда, 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:

$$mv_i = \langle \{atr_k\}, \{ev_j\} \rangle, mv_i \in MV, ev_j \in (V \cup E^{eo=false} \cup MV \cup ME^{eo=false}),$$

где mv_i — вершина метаграфа; atr_k — атрибут, ev_j — элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа.

Следовательно, 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:

$$me_i = \langle v_s, v_e, eo, \{atr_k\}, \{ev_j\} \rangle, e_i \in E, eo = true \mid false,$$

где me_i — метаребро метаграфа; v_s — исходная вершина (метавершина) ребра; v_e — конечная вершина (метавершина) ребра; eo — признак направленности метаребра ($eo=true$ — направленное метаребро, $eo=false$ — ненаправленное метаребро); atr_k — атрибут; ev_j — элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа.

Отсюда, 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 ev_j 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, т. е. 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 др., 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», т. е. фрагмент сети, состоящий из вершин и связей, который может выступать как отдельное целое.

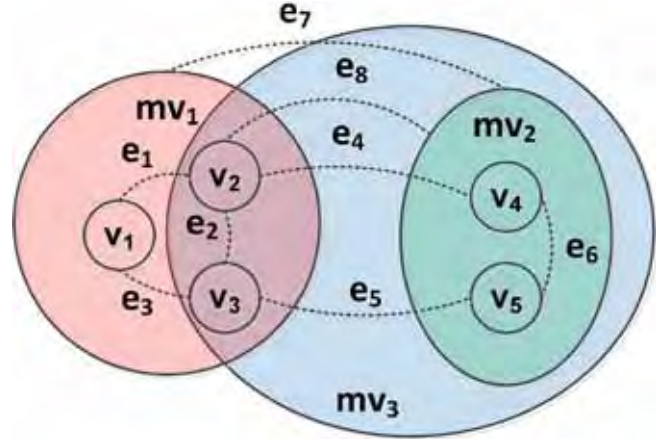


Рис. 2. Пример описания метавершины метаграфа

Пример описания метавершины метаграфа показан на рис. 2. Данный метаграф содержит вершины, метавершины и ребра. На рис. 2 показаны три метавершины: mv_1 , mv_2 и mv_3 . Метавершина mv_1 включает вершины v_1 , v_2 , v_3 и связывающие их ребра e_1 , e_2 , e_3 . Метавершина mv_2 включает вершины v_4 , v_5 и связывающее их ребро e_6 . Ребра e_4 , e_5 являются примерами ребер, соединяющих вершины v_2 - v_4 и v_3 - v_5 , включенные в различные метавершины mv_1 и mv_2 . Ребро e_7 является примером ребра, соединяющего метавершины mv_1 и mv_2 . Ребро e_8 является примером ребра, соединяющего вершину v_2 и метавершину mv_2 . Метавершина mv_3 включает метавершину mv_2 , вершины v_2 , v_3 и ребро e_2 из метавершины mv_1 , а также ребра e_4 , e_5 , e_8 , что показывает холоническую структуру метаграфа.

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

Пример описания метаребра метаграфа представлен на рис. 3. Метаребро содержит метавершины v_s ,

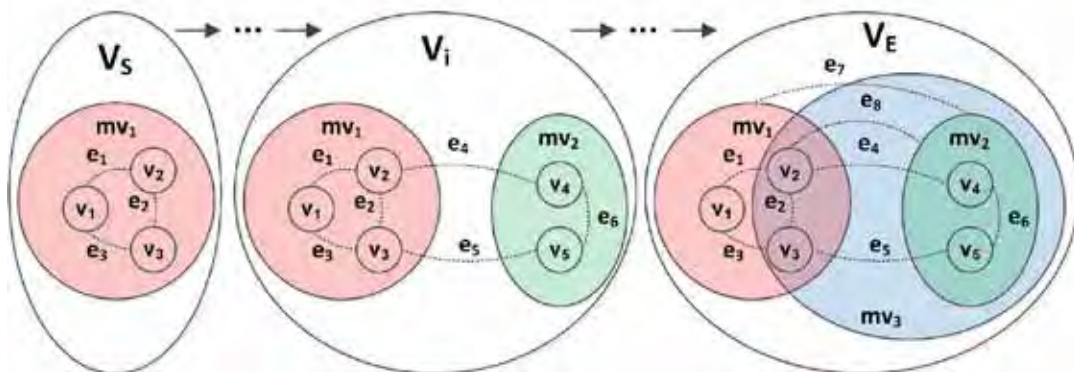


Рис. 3. Пример описания метаребра метаграфа

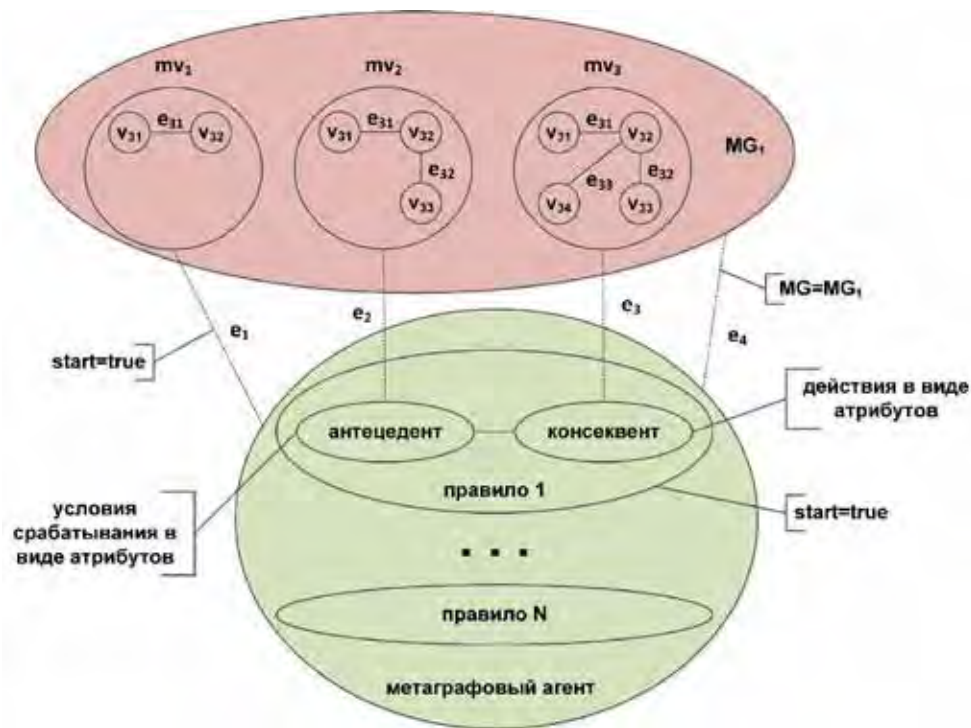


Рис. 4. Представление метаграфового агента в виде фрагмента метаграфа

... v_i , ... v_E и связывающие их ребра. Исходная метавершина содержит фрагмент метаграфа. В процессе преобразования исходной метавершины v_S в конечную метавершину v_E происходит дополнение содержимого метавершины, добавляются новые вершины, связи, вложенные метавершины.

В целом, 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:

$$ag^F = \langle MG_{IN}, MG_{OUT}, AST \rangle,$$

где ag^F — агент-функция; MG_{IN} — метаграф, который выполняет роль входного параметра агента-функции; MG_{OUT} — метаграф, который выполняет роль выходного параметра агента-функции; AST — абстрактное синтаксическое дерево агента-функции, которое можно представить в виде метаграфа.

Определим метаграфовый агент следующим образом:

$$ag^M = \langle MG_D, R, AG^{ST} \rangle, R = \{r_j\},$$

где ag^M — метаграфовый агент; MG_D — метаграф данных и знаний, на основе которого выполняются правила агента; R — набор правил (множество правил r_j); AG^{ST} — стартовое условие выполнения агента (фрагмент метаграфа, который используется для стартовой проверки правил, или стартовое правило).

Структура правила метаграфового агента:

$$r_i : MG_j \rightarrow OP^{MG},$$

где r_i — правило; MG_j — фрагмент метаграфа, на основе которого выполняется правило; OP^{MG} — 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.

Метаграфовый агент представлен в виде метавершины метаграфа. В соответствии с определением, он связан с метаграфом MG_1 , на основе которого выполняются правила агента. Данная связь показана с помощью ребра e_4 .

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 mv_2 , что показано ребром e_2 , а с консеквентом правила связана метавершина данных mv_3 , что показано ребром e_3 . 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

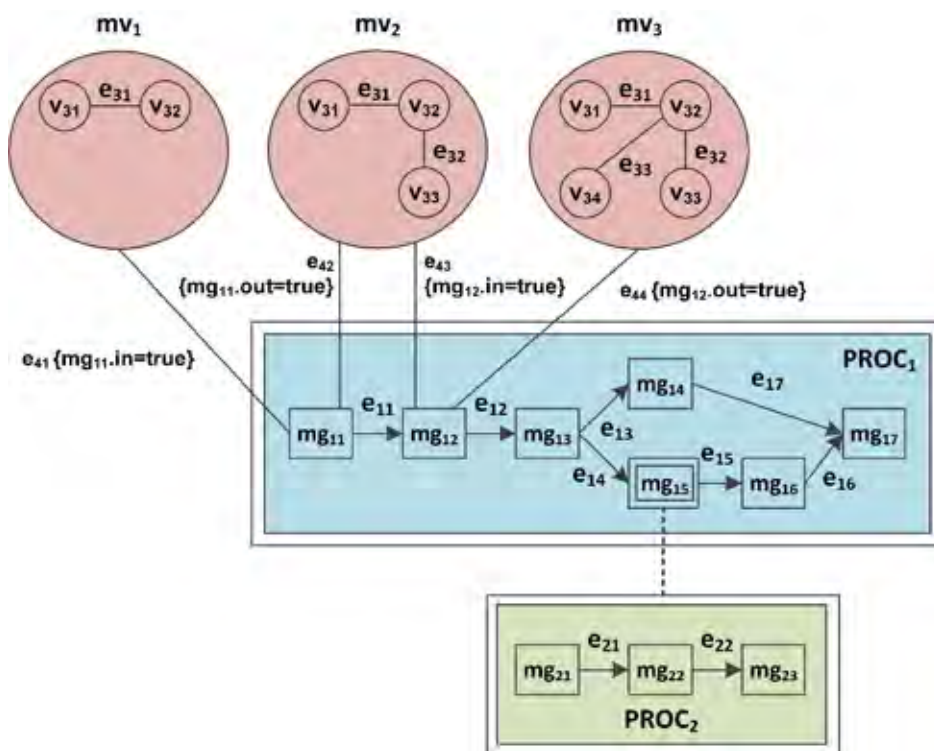


Рис. 5. Пример метаграфового процесса

метаграфа с метавершиной агента, в данном примере это ребро e_1 .

Под «активным узлом метаграфа» mv_{node}^{ACTIVE} будем понимать вершину или метавершину метаграфа, с которой связаны программные агенты, выполняющие определенные функции. Под «метаграфовым процессом» (рис. 5) будем понимать метаребро, базирующееся на активных узлах:

$$PROC_i = \langle v_s, v_e, \{atr_k\}, MG_j [mv_{node} \equiv mv_{node}^{ACTIVE}] \rangle, \\ mv_{node} \in (V \cup MV),$$

где $PROC_i$ — метаграфовый процесс; v_s — исходная вершина (метавершина) процесса; v_e — конечная вершина (метавершина) процесса; atr_k — атрибут; MG_j — фрагмент метаграфа, такой, что каждая его вершина или метавершина mv_{node} является активным узлом метаграфа.

На рис. 5 окружностями показаны вершины и метавершины, используемые для описания данных. Прямоугольниками показаны метавершины или метаребра, соответствующие элементам процесса. Метаграфовые процессы $PROC_1$ и $PROC_2$ 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 v_{i*} (под «*» понимается произвольное значение второго индекса). Элементы процесса соединены направленными ребрами e_{i*} . В качестве примера для элемента процесса v_{15} показаны элементы вложенного подпроцесса. Таким образом, элемент v_{15} одновременно является и эле-

ментом процесса me_1 , и метаребром, которое содержит вложенный процесс.

Данные, поступающие на вход элемента v_{11} , показаны в виде метавершины mv_1 , которая содержит вложенные вершины данных v_{31} и v_{32} и ненаправленную связь между ними e_{31} . Связь метавершины mv_1 с элементом процесса v_{11} осуществляется с помощью ребра e_{41} . Признаком того, что метавершина mv_1 содержит входные данные процесса v_{11} , моделируется с помощью атрибута ребра e_{41} (вершины и ребра могут иметь атрибуты, так как используется модель атрибутивного метаграфа). В данном случае используется атрибут $v_{11}.in=true$. Аналогично с использованием ребер e_{42} , e_{43} , e_{44} производится привязка метавершин mv_2 и mv_3 к элементам процесса v_{12} и v_{12} 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.

Архитектура системы анализа арбитражной практики

Рассмотрим предлагаемую архитектуру информационной системы анализа судебной практики арбитражных судов, построенную на основе подхода ГИИС (рис. 6).

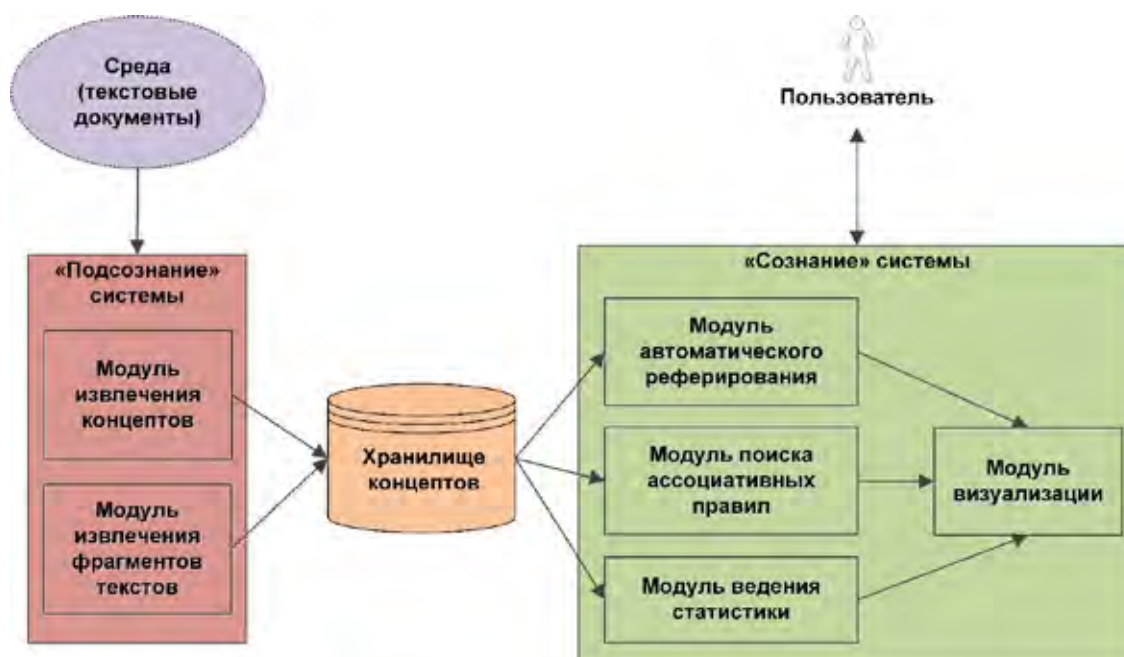


Рис. 6. Архитектура системы анализа арбитражной практики

Классическая архитектура практически любой аналитической информационной системы включает подсистему сбора данных и подсистему обработки данных. В классической архитектуре при проектировании подсистемы сбора рассматриваются в основном 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.

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 от-

ветчика; имя или название заявителя (в жалобах); суть дела; цена иска; решение суда и др.

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 [16]. Для тривиальных случаев используются правила. Например, тип документа, цена иска, решение суда и др. Остальной текст анализируется нейросетевыми моделями на основе фреймворка *TensorFlow* [14].

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 [15].

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 [8] 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 заметить, 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 под сознание 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 др. Для реализации этого модуля может быть использована OLAP-система, в этом случае она может иметь собственное 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

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

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. 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.

Для реализации ГИИС предполагается использовать холоническую многоагентную систему. 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. С использованием метаграфовой модели возможно унифицированное представление архитектуры разрабатываемой системы. Для обработки метаграфовой модели используются два вида агентов: агент-функция и метаграфовый агент. С активным узлом метаграфа связаны программные агенты, выполняющие определенные функции обработки метаграфовой модели. Под «метаграфовым процессом» понимается метаребро, базирующееся на активных узлах.

Рецензент: Сухов Андрей Владимирович, доктор технических наук, профессор, профессор кафедры радиоэлектроники, телекоммуникаций и нанотехнологий Московского авиационного института (национального исследовательского университета), Российская Федерация, г. Москва.
E-mail: avs57@mail.ru

Литература

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

2. Колесников А. В., Кириков И. А., Листопад С. В. Гибридные интеллектуальные системы с самоорганизацией: координация, согласованность, спор. М. : ИПИ РАН, 2014. 189 с.
3. Ловцов Д. А., Ниесов В. А. Модернизация информационной инфраструктуры судопроизводства — ключевое направление оптимизации нагрузки на судебную систему // Российское правосудие. 2014. № 9. С. 30—40.
4. Ловцов Д. А., Ниесов В. А. Актуальные проблемы создания и развития единого информационного пространства судебной системы России // Информационное право. 2013. № 5. С. 13—18.
5. Ловцов Д. А., Сергеев Н. А. Информационно-математическое обеспечение управления безопасностью эргатических систем. III. Экспертная информационная система // НТИ. Сер. 2. Информ. процессы и системы. 2001. № 11. С. 23—30.
6. Прикладные интеллектуальные системы, основанные на мягких вычислениях / Под ред. Н. Г. Ярушкиной. Ульяновск : УлГТУ, 2004. 139 с.
7. 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. Вып. № 1. С. 83—99.
8. Тарасов С. Д. Современные методы автоматического реферирования // Научно-технические ведомости СПбГПУ. 2010. Вып. № 6. С. 59—74.
9. Тарасов В. Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. М. : Эдиториал УРСС, 2002. 352 с.
10. Федосеев С. В. Применение современных технологий больших данных в правовой сфере // Правовая информатика. 2018. № 4. С. 50—58.
11. Черненький В. М., Терехов В. И., Гапанюк Ю. Е. Представление сложных сетей на основе метаграфов // Труды XVIII Всеросс. науч.-техн. конф. «Нейроинформатика-2016». Ч. 1 / НИЯУ МИФИ. М. : МИФИ, 2016. С. 173—178.
12. Черненький В. М., Терехов В. И., Гапанюк Ю. Е. Структура гибридной интеллектуальной информационной системы на основе метаграфов // Нейрокомпьютеры: разработка, применение. 2016. Вып. № 9. С. 3—14.
13. Basu A., Blanning R. Metagraphs and their applications. Springer, 2007. 174 pp.
14. Geron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly, 2019. 856 pp.
15. Kulkarni A., Shivananda A. Natural Language Processing Recipes. Unlocking Text Data with Machine Learning and Deep Learning using Python. Apress, 2019. 234 pp.
16. 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.

THE ARCHITECTURE FOR A HYBRID INTELLIGENT INFORMATION SYSTEM FOR ANALYSING COMMERCIAL COURTS PRACTICE

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

E-mail: gapyu@bmstu.ru

Iurii Gapaniuk, Ph.D. (Technology), Associate Professor at the Bauman Moscow State Technical University, Russian Federation, Moscow.

E-mail: gapyu@bmstu.ru

Keywords: commercial court, hybrid intelligent system, hybrid intelligent information system, information system conscious, information system subconscious, text mining, multiagent system, graph, metagraph, metavertex, metaedge, metagraph process, metagraph agent, active metagraph node.

Abstract.

Purpose of the work: improving the scientific and methodological basis of information support for the commercial courts practice system.

Methods used: methods of hybrid intelligent information systems design based on complex networks models.

Results obtained: an approach to the development of Hybrid Intelligent Information Systems (HIIS) based on the use of modules of conscious, subconscious and communication is considered. An approach for HIIS implementation based on complex networks is studied. It is shown that a metagraph model allows describing data, knowledge, and processes as HIIS

components. The structure of a metagraph agent providing for the processing of the metagraph model is studied, and 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 & knowledge model and metagraph agents. The concept of a metagraph process based on the active metagraph node concept is introduced, and a justification is given for an architecture of a HIIIS for analysing commercial courts practice.

References

1. Kolesnikov A. V. Gibridnye intellektual'nye sistemy. Teoriia i tekhnologiya razrabotki. SPb. : SPbGTU, 2001, 137 pp.
2. Kolesnikov A. V., Kirikov I. A., Listopad S. V. Gibridnye intellektual'nye sistemy s samoorganizatsiei: koordinatsiia, soglasovannost', spor. M. : IPI RAN, 2014, 189 pp.
3. Lovtsov D. A., Niesov V. A. Modernizatsiia informatsionnoi infrastruktury sudoproizvodstva -- klichuevoe napravlenie optimizatsii nagruzki na sudebnuiu sistemu. Rossiiskoe pravosudie, 2014, No. 9, pp. 30-40.
4. Lovtsov D. A., Niesov V. A. Aktual'nye problemy sozdaniia i razvitiia edinogo informatsionnogo prostranstva sudebnoi sistemy Rossii. Informatsionnoe pravo, 2013, No. 5, pp. 13-18.
5. Lovtsov D. A., Sergeev N. A. Informatsionno-matematicheskoe obespechenie upravleniia bezopasnost'iu ergaticheskikh sistem. III. Ekspertnaia informatsionnaia sistema, NTI, ser. 2. Inform. protsessy i sistemy, 2001, No. 11, pp. 23-30.
6. Prikladnye intellektual'nye sistemy, osnovannye na miagkikh vychisleniiakh. Pod red. N. G. Iarushkinoi. Ul'ianovsk : UIGTU, 2004, 139 pp.
7. Samokhvalov E. N., Revunkov G. I., Gapaniuk Iu. E. Ispol'zovanie metagrafov dlia opisaniia semantiki i pragmatiki informatsionnykh sistem. Vestnik MGTU im. N.E. Baumana, ser. "Priborostroenie", 2015, vyp. No. 1, pp. 83-99.
8. Tarasov S. D. Sovremennye metody avtomaticheskogo referirovaniia. Nauchno-tekhnicheskie vedomosti SPbGPU, 2010, vyp. No. 6, pp. 59-74.
9. Tarasov V. B. Ot mnogoagentnykh sistem k intellektual'nym organizatsiiam: filosofia, psikhologiya, informatika. M. : Editorial URSS, 2002, 352 pp.
10. Fedoseev S. V. Primenenie sovremennykh tekhnologii bol'shikh dannykh v pravovoi sfere. Pravovaia informatika, 2018, No. 4, pp. 50-58.
11. Chernen'kii V. M., Terekhov V. I., Gapaniuk Iu. E. Predstavlenie slozhnykh setei na osnove metagrafov. Trudy XVIII Vseross. nauch.-tekhn. konf. "Neuroinformatika-2016", ch. 1, NIIaU MIFI, M. : MIFI, 2016, pp. 173-178.
12. Chernen'kii V. M., Terekhov V. I., Gapaniuk Iu. E. Struktura gibridnoi intellektual'noi informatsionnoi sistemy na osnove metagrafov. Neirokomp'yutery: razrabotka, primeneniye, 2016, vyp. No. 9, pp. 3-14.
13. Basu A., Blanning R. Metagraphs and their applications. Springer, 2007, 174 pp.
14. Geron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly, 2019. 856 pp.
15. Kulkarni A., Shivananda A. Natural Language Processing Recipes. Unlocking Text Data with Machine Learning and Deep Learning using Python. Apress, 2019. 234 pp.
16. 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.