

ПОРІВНЯЛЬНА ТАБЛИЦЯ

із наведенням фрагментів дисертації Слюняєва А. С.
та відповідних фрагментів опублікованих текстів інших авторів
без зазначення авторства

Збіги текстів виділені **жовтим** кольором, перефразування та синоніми – **бірюзовим**, перестановки слів місцями – **зеленим**; твердження особи про те, що це нібито вона щось пропонує, оцінює, розробила чи робить висновки – **фіолетовим**.

Червоним шрифтом поданий коментар щодо фрагментів дисертації Слюняєва А. С.
Через великий обсяг кольором виділено не всі збіги.

№	Фрагменти тексту дисертації, у якій виявлено факти порушення академічної доброчесності	Фрагменти опублікованих текстів інших авторів (без зазначення в дисертації Слюняєва А. С. посилань на джерело)
Вид виявленого порушення: плагіат		
1	<p style="text-align: center;">Слюняєв А. С.</p> <p>Методика побудови інтелектуальної інформаційно-керуючої системи аеропорту на основі мультиагентного підходу. – Дис. ... кандидата технічних наук. – Київ, 2010. (https://nrat.ukrintel.ua/searchdoc/0411U002143/) Нумерація сторінок – як надруковано у файлі pdf на сторінках справа вгорі.</p>	<p style="text-align: center;">Швецов А. Н.</p> <p>Модели и методы построения корпоративных интеллектуальных систем поддержки принятия решений. – Дисс. ... доктора технических наук. – Санкт-Петербург, 2004.</p>
	С. 20.	С. 8.
	<p>Під інформаційно-керуючою системою аеропорту (ІКСА) у роботі розуміється складна територіально-розподілена ієрархічна інформаційна система управління аеропортом, яка переробляє інформаційні потоки і для прийняття рішень в якій необхідний збір та семантичний аналіз інформації від великої кількості різнорідних джерел. У дисертаційній роботі ІКСА розглядається в аспекті обробки інформації та підтримки прийняття рішень.</p> <p>На передній план наукових досліджень висувається завдання підвищення рівня інтелектуальності ІКСА обумовлене протиріччями, які склалися між зростаючою складністю інформаційно-керуючих систем (ІКС) в умовах обмежених можливостей людини щодо прийому та обробки інформації і нездатністю традиційних автоматизованих систем управління (АСУ) подолати цю складність. Вирішення цього завдання дозволить звільнити людину від аналізу деталей і фактів на рівні дрібних подробиць, ручного збору даних і добування знань, підвищити ефективність прийнятих рішень в умовах великого числа змінних станів середовища і обмеженого часу.</p>	<p>КС понимается автором как сложная территориально-распределенная иерархическая система, перерабатывающая информационные, материальные, энергетические, и возможно другие ресурсные потоки, для принятия решений в которой необходим сбор и семантический анализ информации от множества гетерогенных источников. В диссертационной работе КС рассматриваются в аспекте обработки информации и поддержки принятия решений.</p> <p>На передний план научных исследований выдвигается проблема повышения уровня интеллектуальности информационно-управляющих корпоративных систем, вытекающая из противоречий, сложившихся между возрастающей сложностью информационно-управляющих систем (ИУС) в условиях ограниченных возможностей человека по приему и обработке информации, и неспособностью традиционных корпоративных информационных систем (КИС) преодолеть эту сложность, освобождает человека от анализа деталей и фактов на уровне мелких подробностей, принятия решений в условиях большого числа переменных состояний объектов и среды и ограниченного времени, ручного сбора данных и извлечения знаний <...></p>
	С. 20–21.	С. 9.
	<p>Складність сучасних інформаційно-керуючих систем в авіації досягла сьогодні такого рівня, який перевищує можливості людини, задіяної в контурі управління АСУ. Це перевищення людських можливостей виявляється не тільки в обсягах інформації, що підлягають обробці, але і в рівні логічної складності, неможливості аналізу всіх взає-</p>	<p>Сложность современных производственных и корпоративных систем достигла сегодня такого уровня, который превышает возможности человека, задействованного в контуре управления АСУ (КИС). Это превышение человеческих возможностей проявляется не только в объемах информации, подлежащих обработке, но и в уровне логиче-</p>

<p>мозв'язків і відносин.</p> <p>Традиційні ІКС, засновані на стандартній моделі «клієнт – сервер» і традиційних методах управління не забезпечують необхідної якості інформаційного аналізу і необхідної ефективності формування керуючих впливів дій в умовах недостатності апріорної інформації про зовнішнє середовище функціонування, великої кількості різноманітних чинників, нестаціонарного і суб'єктивного їх характеру, змінності цілей і критеріїв якості управління внаслідок деградації або цілеспрямованої реконфігурації системи.</p>	<p>скої складності, неможливості аналізу всіх взаємозв'язків і відносин.</p> <p>Традиционные КИС, основанные на стандартной модели «клиент-сервер» и традиционных методах управления не обеспечивают требуемого качества информационного анализа и необходимой эффективности формирования управляющих воздействий в условиях недостаточности априорной информации о внешней среде функционирования, большого количества трудно учитываемых факторов нестационарности и субъективного их характера, изменчивости целей и критериев качества управления вследствие деградации или целенаправленной реконфигурации.</p>
<p>С. 21.</p>	<p>С. 9.</p>
<p>Проблематика вказаних напрямів досліджень поставлена в роботах таких учених як Вавілов А.А., Варшавський В.І., Віттіх В.А., Герасимов Б.М., Глушков В.М., Гаврилова Т.А., Городецкий В.І., Ларічев О.І., Ліпаєв В.В., Марлей В.Е., Мартін Дж., Мінський М., Песиков Е.Б., Поспелов Г.С., Поспелов Д.А., Попов Е.В., Пупков К.А., Тарасов В.Б., Трахтенгерц Е.А., Хорошевський В.Ф., Шенк Р.І., Швецов А.Н., Яковлев З.А.</p> <p>Таким чином на даний час існує необхідність вирішення актуального наукового завдання, сутність якого полягає у розробці методик побудови інтелектуальної інформаційно-керуючої системи аеропорту, заснованої на методах штучного інтелекту (ШІ) та інженерії знань.</p>	<p>Проблематика указанных направлений исследований поставлена в работах таких отечественных и зарубежных ученых как Вавилов А.А., Варшавский В.И., Виттих В.А., Глушков В.М., Гаврилова Т.А., Городецкий В.И., Евреинов Э.В., Калинин Л.А., Костров А.В., Ларичев О.И., Летичевский А.А., Липаев В.В., Марлей В.Е., Мартин Дж., Минский М., Осипов Г.С., Песиков Э.Б., Поспелов Г.С., Поспелов Д.А., Попов Э.В., Пупков К.А., Советов Б.Я., Стефанюк В.Л., Трахтенгерц Э.А., Фомин Б.Ф., Хорошевский В.Ф., Черноуцкий И.Г., Шенк Р., Ющенко Е.Л., Яковлев С.А., Яшин А.И.</p> <p>Таким образом, сложились условия для исследования теоретических моделей и методов, разработки алгоритмов и программных средств построения корпоративных интеллектуальных систем поддержки принятия решений (КИНС ППР), основанных на методах искусственного интеллекта (ИИ) и инженерии знаний.</p>
<p>С. 22–23.</p>	<p>С. 13.</p>
<p>Методи дослідження. Використано методи системного аналізу для формулювання загальної структури мультиагентної системи управління, математичної логіки для побудови моделей інформаційних об'єктів та інтелектуальних агентів, теорії формальних систем і числень для побудови моделі поведінки інформаційних об'єктів, а також теорія та методи інженерії знань для формування бази знань інтелектуальних компонентів, методи об'єктно-орієнтованого і логічного програмування при розробці моделей взаємодії інтелектуальних агентів.</p>	<p>Методы исследования. Для решения поставленных задач использовались методы системного анализа, математической логики и теории моделей, теория формальных систем и исчислений, теория и методы инженерии знаний, методы объектно-ориентированного и логического программирования, теория формальных языков и грамматик.</p>
<p>С. 24.</p>	<p>С. 13.</p>
<p>2. Створено інженерну методик автоматизованої побудови інтелектуальної інформаційно-керуючої системи, орієнтованої на широкий клас завдань організаційного і технічного управління.</p>	<p>Практическая значимость диссертационной работы состоит в создании инженерных методик автоматизированного построения корпоративных интеллектуальных систем поддержки принятия решений, ориентированных на широкий класс задач организационного и технического управления.</p>
<p>С. 32.</p>	<p>С. 19.</p>
<p>Збільшення обсягу інформації, яку необхідно обробляти особам, що приймають рішення (ОПР), зростання складності вирішуваних завдань в умовах необхідності обліку великого числа взаємозв'язаних чинників і високої динаміки зовнішнього середовища, призвели до появи розподілених інформаційно-керуючих систем (РІКС), які складаються з локальних ІКС, що функціонують у вузлах обчислювальної мережі (ОМ) і спільно вирішують загальні проблеми на основі обміну інформацією і знаннями, здійснюваного як між ІКС, так і між інши-</p>	<p>Увеличение объема информации, которую необходимо обрабатывать лицам, принимающим решения (ЛПР), возрастание сложности решаемых задач в условиях необходимости учета большого числа взаимосвязанных факторов и высокой динамики внешней среды, привели к появлению класса систем поддержки принятия решений (СППР). <...></p> <p><...></p> <p>Специфика функционирования КС все в большей мере требует применения распределенных СППР (РСППР), которые состоят из локальных</p>

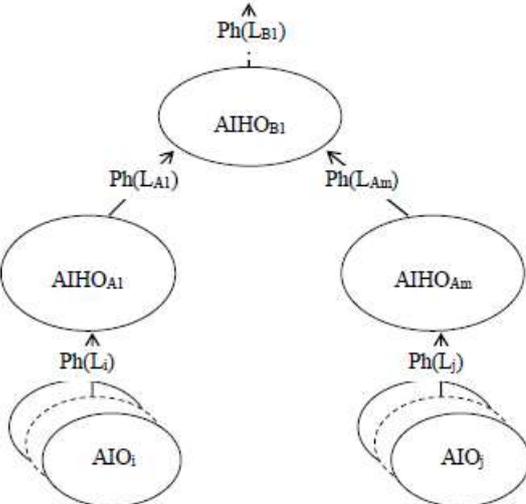
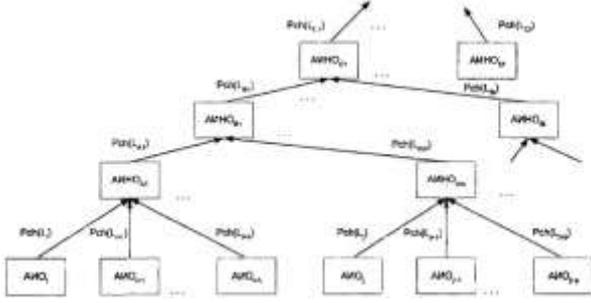
ми підсистемами ІС і розподіленими програмними засобами.	СППР, функционирующих в узлах корпоративной вычислительной сети (ВС), и совместно решающих общие проблемы на основе обмена информацией и знаниями, осуществляемого как между СППР, так и между СППР и другими подсистемами КИС и распределенными приложениями.
С. 32.	С. 21.
<p>У теоретико-множинних позначеннях ІС, що розглядається в аспекті інформаційного обміну, представляється таким чином:</p> $AS = (AIMS, IS, CI), (1.1)$ <p>де AIMS – ІКСА, IS – інформаційне середовище, CI – відображення, що визначає зв'язки ІКСА і середовища (при врахуванні тільки структури зв'язків $CI = AIMS \times IS : \rightarrow \{0, 1\}$).</p> $AIMS = \{IMS_i \mid i=1, \dots, k\}, (1.2)$ <p>де IMS_i – множина локальних ІС, пов'язаних з i-м рівнем управління в ІКС, у свою чергу – $IMS_i = \{IMS_j \mid j = 1, \dots, m\}$, де IMS_j – конкретна локальна ІС, k та m – скінченні натуральні числа.</p>	<p>В теоретико-множественных обозначениях КС, рассматриваемая в аспекте принятия решений, представляется следующим образом:</p> $CS = (CDSS, IS, CI),$ <p>где CDSS - корпоративная СППР, IS - гетерогенная информационная среда, CI - отображение, определяющее связи КСППР и среды (при учете только структуры связей $CI = CDSS \times IS : \rightarrow \{0, 1\}$).</p> $CDSS = \{SetDSS_i \mid i = 1, \dots, k\}$ <p>где SetDSS_i - множество локальных СППР, связанных с i-м уровнем управления в КС, в свою очередь - SetDSS_j - {DSS_j j = 1, ..., m}, где DSS_j - конкретная локальная СППР, k и m - конечные натуральные числа.</p>
С. 33.	С. 21.
$IS = \{IO_i \mid i = 1, \dots, k\}, (1.3)$ <p>де IO_j – множина агентів інформаційного обміну (АІО), пов'язаних з k-м рівнем управління в ІС.</p> $IO_i = \{Se_j \mid j = 1, \dots, \tau\}, (1.4)$ <p>де τ – множина типів АІО, пов'язаних з ІКС. Se_j = {IO_q q = 1, ..., n}, де IO_q – q-й АІО, k та τ – скінченні натуральні числа, n – натуральне число. Множини Se_j – можуть бути зліченими нескінченними множинами.</p> $IO_q = (So, VI, Le), (1.4)$ <p>де So – тип АІО, VI = {Li i = 1, ..., n}, множина мов інформаційного обміну, з якими може працювати даний АІО, Le = {Le_j j = 2, ..., 7} – рівні взаємодії відкритих систем, на яких може здійснювати обмін даний АІО.</p>	$IS = \{SetIO_i \mid i = 1, \dots, k\},$ <p>где SetIO_i - множество АІО, связанных с k-м уровнем управления в КС.</p> $SetIO_i = \{Set_j \mid j = 1, \dots, \tau\},$ <p>где τ - множество сортов АІО, связанных с КС. Set_j - {IO_q q = 1, ..., n}, где IO_q - q-й АІО, k и τ - конечные натуральные числа, n - натуральное число. Множества Set_j - могут быть счетными бесконечными множествами.</p> $IO_q = (Sort, VL, LE),$ <p>где Sort- сорт АІО, VL = {Li i = 1, ..., τ}, множество языков информационного обмена, с которыми может работать данный АІО, LE = {Le_j j = 2, ..., 7} - уровни взаимодействия открытых систем, на которых может осуществлять обмен данный АІО.</p>
С. 33–34.	С. 23.
<p>У формалізованому вигляді модель інформаційної взаємодії в ІКСА визначається таким чином. Розглянемо сукупність АІО, тобто технічних систем і об'єктів, а також суб'єктів (персонал аеропорту), що беруть участь в інформаційних взаємодіях. На відміну від класичного статистичного інформаційного підходу [31–32], що не розглядає структуру і зміст повідомлень, які передаються, в ІС агенти інформаційного обміну повинні отримувати повідомлення певної структури і змісту. Тому, для кожного АІО встановлюється наступна множина повідомлень: M_c^{in} – детермінована множина повідомлень, що приймаються, тобто множина повідомлень які повинні прийматися даним АІО у відомі моменти часу, згідно режимам роботи, регламентам, розкладам та ін.; M_v^{in} – недетермінована множина повідомлень, що приймаються, тобто множина повідомлень які можуть прийматися даним АІО в заздалегідь невідомі моменти часу; M_c^{out} – детермінована множина повідомлень, які передаються і M_v^{out} – недетермінована множина повідомлень, які передаються.</p>	<p>В формальной постановке задача информационной интеграции в КС определяется следующим образом. Рассмотрим совокупность АІО, то есть технических систем и объектов, а также субъектов (персонал корпорации), участвующих в информационных взаимодействиях. В отличие от классического статистического информационного подхода [38], [39], не рассматривающего структуру и содержание передаваемых сообщений, в КС агенты информационного обмена должны получать сообщения определенной структуры и содержания (документы определенных типов, установленные виды отчетов, таблиц, экранных форм и т.п.). Поэтому, для каждого АІО устанавливаются следующие множества сообщений: Mes_c^{in} - множество принимаемых сообщений детерминированное, т.е. множество сообщений которые должны приниматься данным АІО в известные моменты времени, согласно режимам работа, регламентам, расписаниям и т.п.; Mes_v^{in} - множество принимаемых сообщений недетерминированное, т.е. множество сообщений которые могут приниматься данным АІО в заранее неизвестные моменты времени; Mes_c^{out} - множество передаваемых сообщений детерминированное и Mes_v^{out} - множество передаваемых сообщений недетерминированное.</p>
С. 34.	С. 23.

<p>Процеси комунікації в ІС повинні відповідати певним показникам якості, які відповідно до [33], слід задати відповідною матрицею $Q = q_{ij}$, елемент якої q_{ij} є вектор граничних значень показників якості при передачі/прийомі i-го повідомлення з j-м пріоритетом.</p> <p>Тоді інформаційну взаємодію в системі S можна визначити у наступному вигляді:</p> $I_{inf}(S) \Leftrightarrow \left(\begin{array}{l} \forall IO_i (IO_i \in IO) \rightarrow \exists (M_i^{In}(T_i^{In}) \wedge Q_i^{In}) \wedge \\ \wedge (M_i^{Out}(T_i^{Out}) \wedge Q_i^{Out}) \wedge (M_i^{In} \wedge Q_i^{In}) \wedge (M_i^{Out} \wedge Q_i^{Out}) \end{array} \right) \quad (1.5)$ <p>Відображення, що формує часову послідовність передачі/прийому повідомлень</p> $Time(M_{\Psi}^{\xi}, t) = M_{\Psi}^{\xi} \times t \rightarrow T_{\Psi}^{\xi}, \text{ де } t - \text{ поточний час.}$	<p>Процессы коммуникации в КС должны отвечать определенным показателям качества, которые в соответствии с [7], следует задать соответствующей матрицей $Q = q_{ij}$, элемент которой q_{ij} есть вектор граничных значений показателей качества при передаче/приеме i-го сообщения с j-м приоритетом.</p> <p>Тогда информационную интеграцию в системе S можно определить следующей аксиомой существования:</p> $Information_Integration(S) \Leftrightarrow (\forall IO_i (IO_i \in SetIO) \rightarrow \exists (Mes_i^{In}(T_i^{In}) \& Q_i^{In}) \& (Mes_i^{Out}(T_i^{Out}) \& Q_i^{Out}) \& (Mes_i^{In} \& Q_i^{In}) \& (Mes_i^{Out} \& Q_i^{Out})) \quad (1.1)$ <p>Отображение, формирующее временную последовательность передачи/приема сообщений</p> $Time(Mes_{\Psi}^{\xi}, t) = Mes_{\Psi}^{\xi} \times t \rightarrow T_{\Psi}^{\xi}, \text{ где } t - \text{ текущее время.}$
<p>С. 34.</p>	<p>С. 25–26.</p>
<p>Програмно-апаратною платформою інформаційної взаємодії в ІКСА є розподілена об'єктна архітектура. До теперішнього часу сформувався наступні основні технології створення розподілених об'єктних систем:</p> <p>“Загальна архітектура брокера об'єктних запитів” (Common object request Broker Architecture, CORBA) [34–40];</p> <p>система викликів віддалених методів Java (Remove Method Invocation, RMI) [41–44];</p> <p>модель розподілених компонентних об'єктів Microsoft (Distributed Component Object Model, DCOM) [45];</p> <p>технологія Web-служб [46].</p>	<p>Программно-апаратной платформой информационной интеграции в КС являются распределенные объектные архитектуры. КИС, включающая множество взаимосвязанных удаленных программных объектов, называется распределенной объектной системой [57], [61]—[62]. К настоящему времени сформировались следующие основные технологии создания распределенных объектных систем: «Общая архитектура брокера объектных запросов» (Common object request Broker Architecture, CORBA) [63] - [69]; система вызовов удаленных методов Java (Remove Method Invocation, RMI) [70]-[73]; модель распределенных компонентных объектов Microsoft (Distributed Component Object Model, DCOM) [62]; технология Web-служб [74].</p>
<p>С. 34–35.</p>	<p>С. 27.</p>
<p>Основні недоліки відомих моделей і підходів до інформаційної взаємодії полягають в наступному:</p> <p>недостатність ресурсів ІКСА для інтелектуального аналізу і оперативного представлення інформації для прийняття рішень;</p> <p>необхідність розробки великої кількості інтерфейсів і моделей узгодженого подання інформації на рівні форматів даних і програмно-технічної реалізації;</p> <p>не враховується вплив імовірнісних кількісних і часових характеристик інформаційних процесів в умовах ІКСА;</p> <p>обмежені можливості структурно-логічного опису процесів прийняття рішень в ІКСА.</p>	<p>Основные недостатки известных моделей и подходов к информационной интеграции заключаются в следующем:</p> <p>- недостаточность ресурсов КИС для интеллектуального анализа и оперативного представления информации для принятия решений;</p> <p>- необходимость разработки большого количества интерфейсов и моделей согласованного представления информации на уровне форматом данных и программно-технической реализации;</p> <p>- не учитывается влияние вероятностных количественных и временных характеристик информационных процессов в условиях КС;</p> <p>- ограничены возможности структурно-логического описания процессов принятия решений в КС.</p>
<p>С. 35.</p>	<p>С. 27.</p>
<p>Аналіз літератури [41–44] дозволяє зробити наступні висновки:</p> <p>для сучасних ІКСА важливе місце займає проблема інформаційної взаємодії;</p> <p>на теперішній час сформувався дві основні моделі інформаційної взаємодії: об'єднання джерел оперативних даних в структурі однієї ІКС аеропорту і центру узгодження існуючих інформаційних потоків на основі сховища даних;</p> <p>програмно-технічна реалізація інтегрованих ІКС може здійснюватися на основі розподіленої об'єктної архітектури (CORBA, DCOM, Java RMI, Web-сервіси).</p>	<p>На основании анализа, выполненного в данном разделе, можно сделать следующие выводы:</p> <p>- для современных КС важное место занимает проблема информационной интеграции;</p> <p>- сформировались две основных модели информационной интеграции: объединения источников оперативных данных в структуре одной КИС масштаба корпорации и центра согласования существующих информационных потоков на основе ХД;</p> <p>- определенные методологические подходы к информационной; интеграции, языковые модели описания и инструментальные средства разработаны на основе стандартов IDEF;</p> <p>- программно-техническая реализация интегрированных КИС может осуществляться на основе распределенных объектных архитектур (CORBA,</p>

		DCOM, Java RMI, Web-services).
C. 35–36.	<p>Обґрунтування необхідності інтелектуалізації інформаційно-керуючої системи аеропорту.</p> <p>На сьогоднішній день на передній план наукових досліджень висувається завдання підвищення рівня інтелектуальності спеціалізованих керуючих систем, яка витікає з обмеженості традиційного інформаційного підходу.</p> <p>Розглянемо функціонування одного вузла РІКС, обслуговуючого одну ОПР. Хай в середньому для вирішення одного завдання прийняття рішень ОПР необхідно отримати t наборів (множин) параметрів, кожен з яких доставляється одним пакетом даних при середньому обсязі цього пакету q біт. Якщо середня швидкість передачі інформації каналом зв'язку даного вузла РІКС є ϑ (біт/сек), то середній час доставки (отримання) одного пакету параметрів $tq = q / \vartheta$. Час отримання m пакетів для вирішення однієї ЗПР – $tm = mq / \vartheta$.</p> <p>Слюняєв помилково скопіював російську фразу «m наборів» як «t наборів». Недолугий плагіат.</p>	<p>C. 28.</p> <p>1.1.2. Проблема підвищення інтелектуального рівня корпоративних систем підтримки прийняття рішень</p> <p>Сложившаяся к настоящему времени концепция информационной интеграции КИС и СППР приближается к пределу своего естественного развития, за которым на передний план научных исследований выдвигается проблема повышения уровня интеллектуальности информационно-управляющих корпоративных систем, вытекающая из ограниченности традиционного информационного подхода.</p> <p>Рассмотрим функционирование одного узла РСППР, обслуживающего одно ЛПР. Пусть в среднем для решения одной задачи принятия решений ЛПР необходимо получить m наборов (множеств) параметров, каждый из которых доставляется одним пакетом данных при среднем объеме этого пакета q бит. Если средняя скорость передачи информации по каналу связи данного узла РСППР есть ϑ (бит/сек), то среднее время доставки (получения) одного пакета параметров $tq = q / \vartheta$. Время получения m пакетов для решения одной ЗПР – $tm = mq / \vartheta$.</p>
C. 36.	<p>Нехай продуктивність технічних засобів вузла РІКС є W – швидкість обробки інформації в одиницю часу (біт/сек), тоді середній час, що витрачається на вирішення однієї задачі є $t_i = tm + tq + m(tf + td) + tok$, де tQ – час, необхідний на обробку m наборів параметрів технічними засобами вузла, tf – середній час формування запиту на отримання пакету (набору параметрів); td – середній час пошуку відповіді на запит в БД або сховищі; to – середній час на обдумування відповіді в процесі діалогу вузла РІКС і ОПР; k – число кроків діалогу або етапів обдумування. Час tm, tQ, tf, td, to може частково перекриватися, зменшуючи t_i, але це залежить від багатьох чинників, таких як алгоритми роботи конкретного вузла РІКС, стиль роботи ОПР, процедури обробки транзакцій, прийняті в ІКС та ін. Позначаючи, $V = tq$, отримаємо</p> $t_i = V(1/\vartheta + 1/W) + m(t_f + t_d) + t_{ok} \quad (1.6)$	<p>C. 28.</p> <p>Пусть производительность технических средств узла РСППР есть W – число обрабатываемых единиц информации в единицу времени (бит/сек), тогда среднее время, затрачиваемое на решение одной ЗПР есть $t_i = tm + tq + m(tf + td) + tok$, где tQ - время необходимое на обработку m наборов параметров техническими средствами узла, tf - среднее время формировать запроса на получение пакета (набора параметров), td - среднее время поиска ответа на запрос в БД или хранилище, to – среднее время на обдумывание ответа в процессе диалога узла РСППР и ЛПР, k - число шагов диалога или этапов обдумывания. Времена tm, tq, tf, td, to могут частично перекрываться, уменьшая t_i, но это зависит от многих факторов, таких как алгоритмы работы конкретного узла РСППР, поведение и предпочтения ЛПР, процедуры обработки транзакций, принятие в КС и др. Обозначая, $V = tq$, получим</p> $t_i = V(1/\vartheta + 1/W) + m(t_f + t_d) + t_{ok}$
C. 36.	<p>Отже, середнє можливе число завдань, вирішуваних ОПР протягом робочого дня складе $M_c = (T_c - T_r)/t_i$ де T_c – тривалість періоду (робочого дня), T_r – час регламентованих перерв.</p> <p>Час обдумування за різними оцінками коливається від 10 до 35 сек. [47–49].</p> <p>За даними джерел [50–51] для компаній, що мають чисельність до 5 тис. співробітників для одного менеджера (ОПР) є потреба в рішенні від 50 до 200 завдань прийняття рішень на тиждень.</p> <p>Слюняєв пише «Отже», нібито це він робить якийсь висновок, а насправді переписує росій-</p>	<p>C. 28–29.</p> <p>Следовательно, среднее возможное число задач, решаемых ЛПР в течение рабочего дня составит $M_c = (T_c - TR)/t_i$, где T_c - продолжительность рабочего дня, TR - время регламентированных перерывов.</p> <p>Время обдумывания по разным оценкам колеблется от 10 до 35 сек. [76]-[78].</p> <p>По данным источников [79], [80] для компаний, имеющих численность до 5 тыс. сотрудников для одного менеджера (ЛПР) имеется потребность в решении от 50 до 200 ЗПР в неделю.</p>

	ську дисертацію. Плагіат.	
С. 37.	Таким чином, можна зробити висновок про необхідність підвищення інтелектуального рівня ІКС, яке полягає в перекладенні низки завдань прийняття рішень на програмні засоби і зняття інформаційного навантаження з ОПР.	С. 29. Таким образом, можно сделать вывод о необходимости повышения интеллектуального уровня КСППР, которое заключается в переложении ряда задач принятия решений на программные средства и снятия информационной нагрузки с ЛПР. Количественная оценка интеллектуального уровня КСППР устанавливается автором далее в разделе 1.3.
С. 37.	Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.	С. 34.
С. 37.	Побудова інтелектуальних моделей управління і підтримки прийняття рішень стала вже універсальним завданням, не залежним від розміру організації і профілю її діяльності [52–54].	Построение интеллектуальных моделей управления и ППР становится универсальной задачей, не зависящей от размера компании и профиля ее деятельности [89], [91], [92].
С. 37.	На підставі проведеного аналізу, що показує зростаючу необхідність інтелектуальної взаємодії в ІКС визначимо більш точно дане поняття.	С. 35. На основании проведенного анализа, показывающего возрастающую необходимость интеллектуальной интеграции в КСППР, определим более точно данное понятие.
С. 37.	Слюняєв пише «На підставі проведеного аналізу», нібито це він щось там аналізував, а насправді переписує російську дисертацію. Плагіат.	С. 35. Полагаем, что интеллектуальная интеграция базируется на информационной интеграции, но при этом АИО передают и принимают необходимые знания, выраженные в форме совокупностей утверждений (теорем) некоторой формальной теории (нескольких теорий в гетерогенной КС). Очевидно, что не всем АИО будут необходимы сообщения знаний, так как в КС останутся простые исполнительные агенты, работающие только в рамках информационного обмена. Поэтому из множества SetIO выделяем множество агентов интеллектуального обмена (АИНО) - SetIA, для которых устанавливаем: KM_c^{In} - множество принимаемых сообщений знаний детерминированное, KM_v^{In} - множество принимаемых сообщений знаний недетерминированное, KM_c^{Out} - множество передаваемых сообщений знаний детерминированное, KM_v^{Out} - множество передаваемых сообщений знаний недетерминированное, $KM_{\psi}^{\xi} = \{KM_i KM_i \{f_j f_j \in FT_k\}\}$, де FT – деяка формальна теорія. За аналогією з моделлю інформаційної взаємодії визначимо матриці показників якості інтелектуального обміну - $\ KQ\ = [kq_{ij}]$, де kq_{ij} – вектор граничних значень показників якості при передачі i-го повідомлення знань j-м пріоритетом.
С. 37–38.	Тоді аксіома існування інтелектуальної взаємодії в ІКС є	С. 35. Тогда аксиома существования интеллектуальной интеграции в КС есть
	$I_{Inf}(S) \Leftrightarrow \left(\begin{array}{l} I_{Inf}(S) \wedge \forall IA_i (IA_i \in IA) \rightarrow \exists (KM_c^{In}(T_c^{In}) \wedge KQ_c^{In}) \wedge \\ \wedge (KM_c^{Out}(T_c^{Out}) \wedge KQ_c^{Out}) \wedge (KM_v^{In} \wedge KQ_v^{In}) \wedge \\ \wedge (KM_v^{Out} \wedge KQ_v^{Out}) \wedge Cost(I_{Inf} \rightarrow \min) \end{array} \right) \quad (1.7)$	$Intelligent_Integration(S) \Leftrightarrow Information_Integration(S) \wedge (\forall IA_i (IA_i \in SetIA) \rightarrow \exists (KM_c^{In}(T_c^{In}) \wedge KQ_c^{In}) \wedge (KM_c^{Out}(T_c^{Out}) \wedge KQ_c^{Out}) \wedge (KM_v^{In} \wedge KQ_v^{In}) \wedge (KM_v^{Out} \wedge KQ_v^{Out}) \wedge Cost(Information_Integration \rightarrow \min)) \quad (1.2)$

<p>Слюняєв переписав чужу формулу, але замість знака & використав Λ. Плагіат.</p>	
<p>С. 38.</p>	<p>С. 36.</p>
<p>Агенти інформаційного обміну, якими можуть служити розподілені об'єктні додатки, підсистеми ІКСА, аналітичні додатки і підпорядковані системи, передають інформацію до БД або інших сховищ в різних форматах. Повідомлення даних, що передаються АІО, розглядатимемо як речення відповідної формальної мови.</p> <p>У більш загальному випадку припускаємо, що в ІКСА існує кінцева множина різних формальних контекстно-вільних мов $LS = \{L_i i = 1, \dots, k\}$. Множина повідомлень, передаваних АІО, розглядається як сукупність речень у відповідній мові $M^{Out}(IO_i) = \{Pch_j(L_i)\}$ $Pch_j(L_i)$ – j-е речення мови L_i.</p> <p>Слюняєв переписав чужий текст, видаливши покликання. Плагіат.</p>	<p>АІО, котрими можуть служити распределенные объектные приложения, подсистемы КИС (модули ERP-систем), аналитические приложения и унаследованные системы, передают информацию в БД или другие хранилища в различных форматах. Сообщения данных, передаваемые АІО, будем рассматривать как предложения соответствующего формального языка L_i. Случай реляционного языка для АІО рассмотрен в работах [103], [104].</p> <p>В более общем случае предполагаем, что в КС существует конечное множество различных формальных контекстно-свободных языков $LS = \{L_i i = 1, \dots, k\}$. Множество сообщений, передаваемых АІО, рассматриваем как совокупность предложений в соответствующем языке $Mes^{Out}(IO_i) = \{Pch_j(L_i)\}$, где $Pch_j(L_i)$ – j – е предложение языка L_i.</p>
<p>С. 38.</p>	<p>С. 36.</p>
<p>Для забезпечення інтелектуальної взаємодії АІНО повинен певним чином перетворити інформацію, отриману від АІО, з тим, щоб сформувати $KM_{\Psi}^{\xi} = \{KM_i KM_i, \{f_j f_j \in FT_k\}\}$. Слід виділити три процеси формування повідомлень знань:</p> <ul style="list-style-type: none"> - висхідне формування повідомлень знань (рис. 1.3.); - побудова повідомлень за наслідками логічного виводу по розподілених БЗ; - композиційний процес, що суміщає висхідне формування повідомлень знань і логічний вивід. 	<p>Для обеспечения интеллектуальной интеграции АІНО должен определенным образом преобразовать информацию, полученную от АІО, с тем, чтобы сформировать $KM_{\Psi}^{\xi} = \{KMes_i KMes_i = \{f_j f_j \in FormalTheory_k\}\}$. Следует выделить три процесса формирования сообщений знаний:</p> <ul style="list-style-type: none"> - восходящее формирование сообщений знаний (рис. 1.3.); - построение сообщений по результатам логического вывода по распределенным БЗ; - композиционный процесс, совмещающий восходящее формирование сообщений знаний и логический вывод.
<p>С. 38–39.</p>	<p>С. 36, 39.</p>
<p>АІО надсилають свої повідомлення як пропозиції в мовах: $L_i, L_{i+1}, \dots, L_{i+k}$. АІНО формує з отриманих повідомлень власні повідомлення у формальній мові L_j. Таким чином висхідне формування повідомлень знань слід розділити на три етапи:</p> <ol style="list-style-type: none"> 1. усунення надмірності, яке може розглядатися як скорочення ланцюжка $P_{chj}(L_i)$ за допомогою продукційних правил вигляду $\alpha\beta \dots \varphi \rightarrow \alpha \dots \varphi$; 2. виконання функціональних (алгоритмічних) перетворень над $P_{chj}(L_i)$; 3. формування логічних формул в мові L_j за наслідками логічного аналізу $P_{chj}(L_i)$. <p>Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	<p>АІО посылают свои сообщения как предложения в языках $L_i, L_{i+1}, \dots, L_{i+k}$. АІНО формирует из полученных сообщений собственные сообщения в формальном языке L_j. Таким образом, восходящее формирование сообщений знаний следует разделить на три этапа (рис. 1.4.):</p> <ol style="list-style-type: none"> 1. устранение избыточности, которое может рассматриваться как укорочение цепочки $P_{chj}(L_i)$ с помощью продукционных правил вида $\alpha\beta \dots \varphi \rightarrow \alpha \dots \varphi$; 2. выполнение функциональных (алгоритмических) преобразований над $P_{chj}(L_i)$; 3. формирование логических формул в языке L_j по результатам логического анализа $P_{chj}(L_i)$.
<p>С. 39.</p>	<p>С. 37.</p>

 <p>Рис. 1.3. Висхідне формування повідомлень знань</p> <p>Слюняєв скористався частиною рисунка з російської дисертації, трохи змінивши його. Плагіат.</p>	 <p>Рис. 1.3. Восходящее формирование сообщенных знаний</p>
<p>С. 40.</p>	<p>С. 49.</p>
<p>Мультиагентні системи як необхідний компонент інтелектуалізації ІКС.</p> <p>Інтелектуальний агент (ІА) – це програмний або апаратний об'єкт, що автономно функціонує для досягнення цілей, поставлених перед ним власником або користувачем, володіє певними інтелектуальними здібностями. Рівень цих здібностей, необхідних для досягнення поставлених перед ІА цілей, можна визначити, користуючись класифікацією Д.А. Поспелова [55].</p> <p>Для замкнутих середовищ може бути побудовано кінцевий вичерпний опис і агенти, що функціонують в таких середовищах, можуть володіти повним знанням про середовище і його властивості або отримати цю інформацію в процесі своєї взаємодії з середовищем. Трансформовані середовища можуть змінювати свої характеристики і реакції на дії агентів залежно від тих дій, які агенти здійснюють в середовищі. Вид математичного апарату, що дозволяє описати поведінку агента у відповідному середовищі, і є мірою його інтелектуальної складності (або розумності). Представимо цю класифікацію у вигляді табл. 1.1.</p> <p>Слюняєв переписав чужий текст разом із готовим покликанням. Плагіат.</p>	<p>1.2.2. Мультиагентные системы как необходимый компонент интеллектуализации</p> <p><...></p> <p><...> Отталкиваясь от определенных данных в [125], [161], будем считать, что ИА - это программный или аппаратный объект (сущность), автономно функционирующий для достижения целей, поставленных перед ним владельцем или пользователем, обладающий определенными интеллектуальными способностями. Уровень этих способностей, необходимых для достижения поставленных перед ИА целей, можно определить, пользуясь классификацией Д. А. Поспелова [162].</p> <p>Для замкнутых сред может быть построено конечное исчерпывающее описание, и функционирующие в таких средах агенты могут обладать полным знанием о среде и ее свойствах или получить эту информацию в процессе своего взаимодействия со средой. Трансформируемые среды могут изменять свои характеристики и реакции на действия агентов в зависимости от тех действий, которые агенты совершают в среде. Вид математического аппарата, позволяющего описать поведение агента в соответствующей среде, и является мерой его интеллектуальной сложности (или разумности). Представим эту классификацию в виде табл. 1.1.</p>
<p>С. 40.</p>	<p>С. 50.</p>
<p>Таблица 1.1. Классификация интеллектуальных агентов</p>	<p>Таблица 1.1</p>

	Тип середовища функціонування	Метод математичного опису	Вид інтелектуального агента	Тип середі функціонування	Метод математического описания	Вид ИА
	Замкнуте детерміноване	Автомати грамматики, кінцеві автомати	Автомати агенти	Замкнутая детерминированная	Автоматные грамматика, конечные автоматы	Автоматные агенты
	Замкнуте стохастичне	Імовірнісні автомати	Імовірнісні автоматні агенти	Замкнутая вероятностная	Вероятностные автоматы	Вероятностные автоматные агенты
	Не трансформоване, відкрите	Контекстно-вільні граматики, сценарії, магазинні автомати	КВГ-агенти	Не трансформируемая открытая	Контекстно-свободные грамматика, сценарии, магазинные автоматы	КСГ-агенты
	Трансформоване, замкнуте	Контекстно-залежні граматики (линійні автомати)	КЗГ-агенти	Трансформируемая замкнутая	Контекстно-зависимые грамматика (линейные автоматы)	КЗГ-агенты
	Трансформоване, відкрите	Семантичні системи	Семантичні агенти	Трансформируемая открытая	Семантические системы	Семантические агенты
С. 41.	<p>Дві базові характеристики – автономність і цілеспрямованість – дозволяють відрізнити ІА від інших програмних і апаратних об'єктів (модулів, підпрограм, процедур і тому подібне). Наявність доцільності поведінки вимагає щоб ІА володів властивістю реактивності. Такий рівень інтелекту відповідає рефлексивній поведінці тварини. Якщо ж ІА володіє знаннями про середовище, власні цілі і способи їх досягнення, то такий агент може бути названо розумним (когнітивним). Таким чином, може бути проведена межа між інтелектуальними і неінтелектуальними агентами.</p> <p>Значні зусилля по стандартизації агентних систем і технологій робить FIPA (Foundation for Intelligent Physical Agents) [56], яка є міжнародною організацією, що виконує роботу по розробці відкритих специфікацій, підтримуючих інтероперабельність агентів і агентних застосувань. Згідно специфікації абстрактної архітектури, агент визначається як обчислювальний процес, який реалізує автономну, комунікаційну функціональність додатків [57]. Це визначення само по собі нічого не говорить про інтелектуальні здібності агента.</p> <p>Слюняєв переписав чужий текст разом із готовими покликаннями. Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>			С. 50.		
С. 41.	<p>Якщо підійти до класифікації агентів з позицій розумних властивостей, властивих як живим істотам, так і технічним (штучним) системам, то можна побудувати наступну схему (табл. 1.2), спираючись на матеріали А. Сломана і С. Франкліна [58–59]. Значення введених параметрів можна розглядати як координати вектора в параметричному просторі або як кодове слово виду S1 S2 M1 M2 M3 M4 T1 T2 T3 в трійковій системі числення.</p>			С. 50–51.		
С. 41–42.	<p>До теперішнього часу сформувався досить великий список властивостей, якими повинні володіти ІА [60–62]: автономність, реактивність, активність, базові знання, переконання, бажання, цілі, наміри. Суспільна поведінка агентів може приймати різні форми, які можуть бути класифіковані по рівням взаємодії [63]: зв'язність, координація, кооперація, співпраця, утворення союзу.</p>			<p>Если подойти к классификации агентов с позиций разумных свойств, присущих как живым существам, так и техническим (искусственным) системам, то можно построить следующую схему (табл. 1.2), опираясь на материалы А. Сломана (Aaron Sloman) и С. Франклина (Stan Franklin) [165], [166].</p> <p>Значения введенных параметров можно рассматривать как координаты вектора в параметрическом пространстве или как кодовое слово вида S1 S2 M1 M2 M3 M4 T1 T2 T3 в тричной системе счисления.</p>		
С. 42.	<p>До теперішнього часу сформувався досить великий список властивостей, якими повинні володіти ІА [60–62]: автономність, реактивність, активність, базові знання, переконання, бажання, цілі, наміри. Суспільна поведінка агентів може приймати різні форми, які можуть бути класифіковані по рівням взаємодії [63]: зв'язність, координація, кооперація, співпраця, утворення союзу.</p>			С. 52.		
С. 42.	<p>До теперішнього часу сформувався досить великий список властивостей, якими повинні володіти ІА [60–62]: автономність, реактивність, активність, базові знання, переконання, бажання, цілі, наміри. Суспільна поведінка агентів може приймати різні форми, які можуть бути класифіковані по рівням взаємодії [63]: зв'язність, координація, кооперація, співпраця, утворення союзу.</p>			<p>К настоящему времени сформировался довольно большой список свойств, которыми должны обладать ИА [168]—[170]: автономность, реактивность, активность, базовые знания, убеждения, желания, цели, намерения. Общественное поведение агентов может принимать разные формы, которые могут быть классифицированы по уровням взаимодействия [171]: связность, координация, кооперация, сотрудничество, образование союза.</p>		
С. 42.	<p>До теперішнього часу сформувався досить великий список властивостей, якими повинні володіти ІА [60–62]: автономність, реактивність, активність, базові знання, переконання, бажання, цілі, наміри. Суспільна поведінка агентів може приймати різні форми, які можуть бути класифіковані по рівням взаємодії [63]: зв'язність, координація, кооперація, співпраця, утворення союзу.</p>			С. 51.		

		Таблиця 1.2.		Таблиця 1.2.																																																																																																																							
		Інтелектуальні властивості агентів		Інтелектуальні властивості агентів																																																																																																																							
		<table border="1"> <tr> <td rowspan="2">Сенсорика (відчуття)</td> <td rowspan="2">S1</td> <td>0</td> <td>З одним відчуттям</td> </tr> <tr> <td>1</td> <td>З декількома відчуттями</td> </tr> <tr> <td rowspan="3">Пам'ять</td> <td rowspan="3">M1</td> <td>0</td> <td>Тільки одне з багатьох відчуттів в даний ситуації</td> </tr> <tr> <td>1</td> <td>Множинні відчуття на одиничний об'єкт, подія або ситуацію</td> </tr> <tr> <td>2</td> <td>Множинні відчуття на багатовимірний об'єкт, подія або ситуацію</td> </tr> <tr> <td rowspan="12">Моторика (дія в часі)</td> <td rowspan="2">T1</td> <td>0</td> <td>Без пам'яті</td> </tr> <tr> <td>1</td> <td>З пам'яттю на миттєві події (кінцеві)</td> </tr> <tr> <td rowspan="2">M2</td> <td>0</td> <td>З потенційно необмеженою або нескінченною пам'яттю</td> </tr> <tr> <td>1</td> <td>Тільки з короткочасною пам'яттю</td> </tr> <tr> <td rowspan="2">M3</td> <td>0</td> <td>З короткочасною і довготривалою пам'яттю</td> </tr> <tr> <td>1</td> <td>Не може навчатися (не втрачує довготривалу пам'ять)</td> </tr> <tr> <td rowspan="2">M4</td> <td>0</td> <td>Може навчатися (втрачує довготривалу пам'ять)</td> </tr> <tr> <td>1</td> <td>Зберігає сенсорну інформацію по деякому відчуттю</td> </tr> <tr> <td rowspan="2">T2</td> <td>0</td> <td>Зберігає сенсорну інформацію по всіх відчуттях</td> </tr> <tr> <td>1</td> <td>Діє тільки в реальному часі</td> </tr> <tr> <td rowspan="2">T3</td> <td>0</td> <td>Може планувати дії</td> </tr> <tr> <td>1</td> <td>Не може візуалізувати відчуття</td> </tr> <tr> <td rowspan="2">T4</td> <td>0</td> <td>Може візуалізувати деякі відчуття</td> </tr> <tr> <td>1</td> <td>Може візуалізувати всі відчуття</td> </tr> <tr> <td rowspan="2">T5</td> <td>0</td> <td>Не має моделі середовища існування</td> </tr> <tr> <td>1</td> <td>Має умовну модель</td> </tr> <tr> <td rowspan="2">T6</td> <td>0</td> <td>Може створювати ментальні моделі середовища</td> </tr> <tr> <td>1</td> <td>Може створювати ментальні моделі середовища</td> </tr> </table>	Сенсорика (відчуття)	S1	0	З одним відчуттям	1	З декількома відчуттями	Пам'ять	M1	0	Тільки одне з багатьох відчуттів в даний ситуації	1	Множинні відчуття на одиничний об'єкт, подія або ситуацію	2	Множинні відчуття на багатовимірний об'єкт, подія або ситуацію	Моторика (дія в часі)	T1	0	Без пам'яті	1	З пам'яттю на миттєві події (кінцеві)	M2	0	З потенційно необмеженою або нескінченною пам'яттю	1	Тільки з короткочасною пам'яттю	M3	0	З короткочасною і довготривалою пам'яттю	1	Не може навчатися (не втрачує довготривалу пам'ять)	M4	0	Може навчатися (втрачує довготривалу пам'ять)	1	Зберігає сенсорну інформацію по деякому відчуттю	T2	0	Зберігає сенсорну інформацію по всіх відчуттях	1	Діє тільки в реальному часі	T3	0	Може планувати дії	1	Не може візуалізувати відчуття	T4	0	Може візуалізувати деякі відчуття	1	Може візуалізувати всі відчуття	T5	0	Не має моделі середовища існування	1	Має умовну модель	T6	0	Може створювати ментальні моделі середовища	1	Може створювати ментальні моделі середовища	<table border="1"> <tr> <td rowspan="2">Сенсорика (чутства)</td> <td rowspan="2">S1</td> <td>0</td> <td>С одним чуттями</td> </tr> <tr> <td>1</td> <td>С нескінченними чуттями</td> </tr> <tr> <td rowspan="3">Пам'ять</td> <td rowspan="3">M1</td> <td>0</td> <td>Тільки одне із миттєвих чутств в даний ситуації</td> </tr> <tr> <td>1</td> <td>Множинні чутства на одиничний об'єкт, подія або ситуацію</td> </tr> <tr> <td>2</td> <td>Множинні чутства на багатовимірний об'єкт, подія або ситуацію</td> </tr> <tr> <td rowspan="12">Моторика (дія в часі)</td> <td rowspan="2">T1</td> <td>0</td> <td>Без пам'яті</td> </tr> <tr> <td>1</td> <td>С пам'яттю на прошедшие события (кінцеві)</td> </tr> <tr> <td rowspan="2">M2</td> <td>0</td> <td>С потенційно необмеженою або нескінченною пам'яттю</td> </tr> <tr> <td>1</td> <td>Тільки з короткочасною пам'яттю</td> </tr> <tr> <td rowspan="2">M3</td> <td>0</td> <td>С короткочасною і довготривалою пам'яттю</td> </tr> <tr> <td>1</td> <td>Не може навчатися (не втрачує довготривалу пам'ять)</td> </tr> <tr> <td rowspan="2">M4</td> <td>0</td> <td>Може навчатися (втрачує довготривалу пам'ять)</td> </tr> <tr> <td>1</td> <td>Зберігає сенсорну інформацію по деякому відчуттю</td> </tr> <tr> <td rowspan="2">T2</td> <td>0</td> <td>Зберігає сенсорну інформацію по всіх відчуттях</td> </tr> <tr> <td>1</td> <td>Діє тільки в реальному часі</td> </tr> <tr> <td rowspan="2">T3</td> <td>0</td> <td>Може планувати дії</td> </tr> <tr> <td>1</td> <td>Не може візуалізувати відчуття</td> </tr> <tr> <td rowspan="2">T4</td> <td>0</td> <td>Може візуалізувати деякі відчуття</td> </tr> <tr> <td>1</td> <td>Може візуалізувати всі відчуття</td> </tr> <tr> <td rowspan="2">T5</td> <td>0</td> <td>Не має моделі середовища існування</td> </tr> <tr> <td>1</td> <td>Має умовну модель</td> </tr> <tr> <td rowspan="2">T6</td> <td>0</td> <td>Може створювати ментальні моделі середовища</td> </tr> <tr> <td>1</td> <td>Може створювати ментальні моделі середовища</td> </tr> </table>	Сенсорика (чутства)	S1	0	С одним чуттями	1	С нескінченними чуттями	Пам'ять	M1	0	Тільки одне із миттєвих чутств в даний ситуації	1	Множинні чутства на одиничний об'єкт, подія або ситуацію	2	Множинні чутства на багатовимірний об'єкт, подія або ситуацію	Моторика (дія в часі)	T1	0	Без пам'яті	1	С пам'яттю на прошедшие события (кінцеві)	M2	0	С потенційно необмеженою або нескінченною пам'яттю	1	Тільки з короткочасною пам'яттю	M3	0	С короткочасною і довготривалою пам'яттю	1	Не може навчатися (не втрачує довготривалу пам'ять)	M4	0	Може навчатися (втрачує довготривалу пам'ять)	1	Зберігає сенсорну інформацію по деякому відчуттю	T2	0	Зберігає сенсорну інформацію по всіх відчуттях	1	Діє тільки в реальному часі	T3	0	Може планувати дії	1	Не може візуалізувати відчуття	T4	0	Може візуалізувати деякі відчуття	1	Може візуалізувати всі відчуття	T5	0	Не має моделі середовища існування	1	Має умовну модель	T6	0	Може створювати ментальні моделі середовища	1	Може створювати ментальні моделі середовища
Сенсорика (відчуття)	S1	0			З одним відчуттям																																																																																																																						
		1	З декількома відчуттями																																																																																																																								
Пам'ять	M1	0	Тільки одне з багатьох відчуттів в даний ситуації																																																																																																																								
		1	Множинні відчуття на одиничний об'єкт, подія або ситуацію																																																																																																																								
		2	Множинні відчуття на багатовимірний об'єкт, подія або ситуацію																																																																																																																								
Моторика (дія в часі)	T1	0	Без пам'яті																																																																																																																								
		1	З пам'яттю на миттєві події (кінцеві)																																																																																																																								
	M2	0	З потенційно необмеженою або нескінченною пам'яттю																																																																																																																								
		1	Тільки з короткочасною пам'яттю																																																																																																																								
	M3	0	З короткочасною і довготривалою пам'яттю																																																																																																																								
		1	Не може навчатися (не втрачує довготривалу пам'ять)																																																																																																																								
	M4	0	Може навчатися (втрачує довготривалу пам'ять)																																																																																																																								
		1	Зберігає сенсорну інформацію по деякому відчуттю																																																																																																																								
	T2	0	Зберігає сенсорну інформацію по всіх відчуттях																																																																																																																								
		1	Діє тільки в реальному часі																																																																																																																								
	T3	0	Може планувати дії																																																																																																																								
		1	Не може візуалізувати відчуття																																																																																																																								
T4	0	Може візуалізувати деякі відчуття																																																																																																																									
	1	Може візуалізувати всі відчуття																																																																																																																									
T5	0	Не має моделі середовища існування																																																																																																																									
	1	Має умовну модель																																																																																																																									
T6	0	Може створювати ментальні моделі середовища																																																																																																																									
	1	Може створювати ментальні моделі середовища																																																																																																																									
Сенсорика (чутства)	S1	0	С одним чуттями																																																																																																																								
		1	С нескінченними чуттями																																																																																																																								
Пам'ять	M1	0	Тільки одне із миттєвих чутств в даний ситуації																																																																																																																								
		1	Множинні чутства на одиничний об'єкт, подія або ситуацію																																																																																																																								
		2	Множинні чутства на багатовимірний об'єкт, подія або ситуацію																																																																																																																								
Моторика (дія в часі)	T1	0	Без пам'яті																																																																																																																								
		1	С пам'яттю на прошедшие события (кінцеві)																																																																																																																								
	M2	0	С потенційно необмеженою або нескінченною пам'яттю																																																																																																																								
		1	Тільки з короткочасною пам'яттю																																																																																																																								
	M3	0	С короткочасною і довготривалою пам'яттю																																																																																																																								
		1	Не може навчатися (не втрачує довготривалу пам'ять)																																																																																																																								
	M4	0	Може навчатися (втрачує довготривалу пам'ять)																																																																																																																								
		1	Зберігає сенсорну інформацію по деякому відчуттю																																																																																																																								
	T2	0	Зберігає сенсорну інформацію по всіх відчуттях																																																																																																																								
		1	Діє тільки в реальному часі																																																																																																																								
	T3	0	Може планувати дії																																																																																																																								
		1	Не може візуалізувати відчуття																																																																																																																								
T4	0	Може візуалізувати деякі відчуття																																																																																																																									
	1	Може візуалізувати всі відчуття																																																																																																																									
T5	0	Не має моделі середовища існування																																																																																																																									
	1	Має умовну модель																																																																																																																									
T6	0	Може створювати ментальні моделі середовища																																																																																																																									
	1	Може створювати ментальні моделі середовища																																																																																																																									
	C. 42–43.	C. 52.																																																																																																																									
	<p>Базові знання є необхідним традиційним компонентом для всіх ІНС, переконання ж повинні бути певним чином інтерпретовані в структурі МАС. Агент може сприймати як дійсні правила формування висновків, базові шкали і ваги критеріїв, так і функції або відношення переваги та ін.</p> <p>Переконання можна розділити на три класи. Внутрішні переконання агента – це алгоритми, сценарії, оцінки, закладені в нього при розробці або внесенні в процесі експлуатації власником або користувачем. Індуктивні переконання, що виникають в результаті аналізу станів середовища, формуються продукційними правилами наступного вигляду: «Якщо спостерігається факт X, то переконання Z». Комунікативні переконання, зв'язки, що виникають в результаті взаємодії з іншими агентами формуються продукційними правилами вигляду «Якщо A повідомляє про факт X і A – заслужовує довіри джерело, то переконання Z».</p>	<p>Базовые знания являются необходимым традиционным компонентом для всех ИНС, убеждения же должны быть определенным образом интерпретированы в структуре МАС. Агент может воспринимать как истинные правила формирования выводов, базовые шкалы и веса критериев, функции или отношения предпочтения и т. д.</p> <p>Убеждения можно разделить на три класса. Внутренние убеждения агента – это алгоритмы, сценарии, оценки, заложенные в него при разработке или внесенные в процессе эксплуатации владельцем или пользователем. Индуктивные убеждения, возникающие в результате анализа состояния среды, формируются продукционными правилами следующего вида: «Если наблюдается факт X, то убеждение Z». Коммуникативные убеждения, возникающие в результате связи с другими агентами, формируются продукционными правилами вида «Если A сообщает о факте X и A – заслуживающий доверия источник, то убеждение Z».</p>																																																																																																																									
	C. 43.	C. 53.																																																																																																																									
	<p>Велика кількість науково-дослідних лабораторій, університетів, різних фірм і промислових організацій працюють в області агентних систем і технологій. Найбільш відомими дослідницькими центрами є університет Карнегі Меллон, Массачусетський університет, університет м. Болоньї, низка університетів і коледжів Великобританії, Московський державний університет (Росія). Займаються цими проблемами і потужні корпорації (IBM, Microsoft, DEC, Apple, Toshiba, Hewlett Packard і ін.).</p>	<p>Большое количество научно-исследовательских лабораторий, университетов, различных фирм и промышленных организаций работают в области агентных систем и технологий. Наиболее известными исследовательскими центрами являются университет Карнеги Мэллон (Carnegie Mellon University), Массачусетский университет (University of Massachusetts at Amherst), университет г. Болонья (Universita di Bologna), ряд университетов и колледжей Великобритании (Stanford University, Manchester Metropolitan University). Занимаются этими проблемами и крупные корпорации (IBM, Microsoft, DEC, Apple, Toshiba, Hewlett Packard и др.).</p>																																																																																																																									
	C. 43.	C. 53.																																																																																																																									
	<p>Основними напрямками наукового пошуку є теорії агентів, які розглядають математичні методи і формалізми абстрактного представлення структури і властивостей агентів і способи побудови міркувань в таких формальних системах; методи колективної поведінки агентів; архітектура агентів і МАС; методи, мови і засоби комунікації агентів; мови програмування агентів; методи і засоби автоматизованого проектування МАС; методи і засоби забезпечення мобільності агентів.</p> <p>Труднощі формального представлення інтенціональних (ментальних) понять, таких як переконання, наміри, бажання, були добре показані М.</p>	<p>Основными направлениями научного поиска являются теории агентов, которые рассматривают математические методы и формализмы абстрактного представления структуры и свойств агентов и способы построения рассуждений в таких формальных системах; методы коллективного поведения агентов; архитектуры агентов и МАС; методы, языки и средства коммуникации агентов; языки программирования агентов; методы и средства автоматизированного проектирования МАС; методы и средства обеспечения мобильности агентов.</p> <p>Трудности формального представления интенциональных (ментальных) понятий, таких как убеждения, намерения, желания, были хорошо показаны М.</p>																																																																																																																									

<p>Вулдріджем та Н.Р. Дженінгсом у [60]. У розробці логічних формалізмів ментальних понять виникають дві основні проблеми: представлення синтаксису мови формалізації і створення його семантичної моделі.</p>	<p>дения, намерения, желания, были хорошо показаны М. У. Wooldridge, N. R. Jennings [168]. В разработке логических формализмов ментальных понятий возникают две основные проблемы: представление синтаксиса языка формализации и создание его семантической модели.</p>
<p>С. 43.</p>	<p>С. 54.</p>
<p>До проблеми опису синтаксису існує два фундаментальні підходи: застосування модальних логік, що містять не істинні модальні оператори; застосування метамов, заснованих на багатосортній логіці першого порядку, що містить терміни, які визначають формули деякого іншої об'єктної мови.</p>	<p>К проблеме описания синтаксиса существует два фундаментальных подхода: - применение модальных логик, содержащих не истинностные модальные операторы; - применение метаязыков, основанных на многосортной логике первого порядка, содержащей термины, которые определяют формулы некоторого другого объектного языка:</p>
<p>С. 44.</p>	<p>С. 54.</p>
<p>Формалізація знань і переконань в модальних логічних системах приводить до того, що агентів повинні бути відомі всі загальнозначущі формули (пропозиціональні тавтології), а оскільки кількість таких формул нескінченна, то такі моделі виявляються непридатними в реальних технічних системах, оскільки всі вони мають обмежені ресурси.</p> <p>Використання метамов дозволяє представити відношення між термінами метамови, що визначає агента і термінами об'єктної мови, що визначає деякі формули. Недоліком формалізму метамов є суперечність деяких з них [64–66], але існують і успішні метамовні конструкції [67–70].</p> <p>Моделі можливих світів для логіки знань і переконань спираються на основоположні роботи Я. Хінтікки і формулювання нормальної модальної логіки, даної С. Кріпке [71–73]. Оскільки такі семантичні теорії агентів приводять до серйозних внутрішніх загальнозначущих проблем, то досліджуються і альтернативні підходи, серед яких слід назвати теорію намірів Коена і Левескю [74–75], модель Рао і Георгієва [76–77], що розглядає три примітивні модальності: переконання, бажання і намір; сімейство логік Сінга для представлення намірів, переконань, знань і комунікацій агентів в мережі з передачею тимчасового управління [78–79].</p>	<p>Формализация знаний и убеждений в модальных логических системах приводит к тому, что агенту должны быть известны все общезначимые формулы (пропозициональные тавтологии), а поскольку количество таких формул бесконечно, то такие модели оказываются неприменимыми в реальных технических системах, так как все они имеют ограниченные ресурсы.</p> <p>Использование метаязыков позволяет представить отношения между терминами метаязыка, определяющего агента и терминами объектного языка, определяющего некоторые формулы. Недостатком формализма метаязыков является противоречивость некоторых из них [172]–[174], но существуют и успешные метаязыковые конструкции [175]–[178].</p> <p>Модели возможных миров для логики знаний и убеждений опираются на основополагающие работы Я. Хинтикки (J. Hintikka) и формулировку нормальной модальной логики, данную Крипке (S. Kripke) [179]–[181]. Поскольку такие семантические теории агентов приводят к серьезным внутренним общезначимым проблемам, то исследуются и альтернативные подходы, среди которых следует назвать теорию намерений Коэна и Левескю (Cohen, Levesque) [182], [183], модель Рао и Георгієва (Rao & Georgeff) [184], [185], рассматривающую три примитивных модальности: убеждение, желание и намерение; семейство логик Сінга (Singh) для представления намерений, убеждений, знаний и коммуникаций агентов в сети с передачей временного управления [186], [187].</p>
<p>С. 44–45.</p>	<p>С. 66.</p>
<p>1.3. Аналіз існуючих підходів до побудови мультиагентних інформаційно-керуючих систем</p> <p>Оскільки у розділі 1.2 була виявлена необхідність інтелектуальної взаємодії ІКСА і в рамках загальної концепції побудови розподіленої інформаційно-керуючої системи аеропорту сформульовано агентно-орієнтований підхід до інтелектуальної взаємодії компонентів, то в даному розділі досліджується питання про те, наскільки існуючі підходи і методи можуть бути застосовані до поставленого завдання щодо розробки інтелектуальної інформаційно-керуючої системи аеропорту.</p> <p>У роботах Рибіної Г.В., Пишагіна С.В., Льовіна Д.Е. [80–83] розроблений набір методів і програмних інструментів (комплекс АТ-Технологія), що дозволяють підтримувати повний життєвий цикл ав-</p>	<p>1.4. Анализ существующих подходов к построению мультиагентных систем поддержки принятия решений</p> <p>Поскольку в разделе 1.1.3. была выявлена проблема интеллектуальной интеграции КС и в разд. 1.3. в рамках концепции КИНС сформулирован агентноориентированный подход к интеллектуальной интеграции компонентов, то в данном разделе исследуется вопрос о том, насколько существующие подходы и методологии применимы к поставленным задачам и соответствуют заявленной концепции.</p> <p>В работах Рыбиной Г.В., Пишагина С.В., Левина Д.Е. [190]–[193] разработан набор методов и программных инструментов (комплекс АТ-ТЕХНОЛОГИЯ), позволяющих поддерживать полный; жизненный цикл автоматизированного построения</p>

<p>томатизованої побудови ІНС для окремих класів вирішуваних завдань і типів проблемних областей на основі конкретної задачно-орієнтованої методології побудови ІНС в статичних і динамічних проблемних областях.</p>	<p>ИИС для отдельных классов решаемых задач и типов проблемных областей на основе конкретной задачно-ориентированной методологии построения ИИС в статических и динамических проблемных областях.</p>
<p>С. 45.</p> <p>В основі такого підходу лежить багаторівнева модель розгляду процесу взаємодії в ІНС, моделювання конкретних типів, завдань, релевантних технологій традиційних ЕС (підхід “від завдання”), методи і способи побудови моделі архітектури ІНС і відповідних програмних компонентів на кожному рівні взаємодії. Це дозволяє достатньо систематизовано і наочно провести аналіз основних типів ІНС (у яких присутні компоненти типу простих ІС) незалежно від специфічних особливостей конкретної проблемної області, узявши за основу тільки принцип, умовно названий “з погляду архітектури”.</p> <p>Але дані методи і інструментальні засоби не орієнтовані на створення агентно-орієнтованих систем і не забезпечують автономного функціонування ІНС, хоча і можуть бути використані як засіб інтелектуальної взаємодії.</p> <p>Методи проектування агентно-орієнтованих систем до теперішнього часу все ще знаходяться у стадії становлення і постановки завдань. Проблематиці цього напрямку присвячені роботи В.А. Віттіха, В.І. Городецького, П.О. Скобелева, М. Вулдріджа, Н.Р. Дженінгса, М. Георгієва, А. Рао, Д. Кіні, Е.А. Кендала, К. Цетнаровича, Е. Цетнаровича, Е. Наварескі і ін. [84–86].</p>	<p>С. 66.</p> <p>Кратко характеризуюя предложенную методологию, следует отметить, что в ее основе лежит многоуровневая модель рассмотрения процесса интеграции в ИИС, моделирование конкретных типов задач, релевантных технологии традиционных ЭС (подход «от задачи»), методы и способы построения модели архитектуры ИИС и соответствующих программных компонентов на каждом уровне интеграции. Это позволяет достаточно систематизированно и наглядно провести анализ основных типов ИИС (в которых присутствуют компоненты типа простых ЭС) независимо от специфических особенностей конкретной проблемной области, взяв за основу только принцип, условно названный «с точки зрения архитектуры».</p> <p>Но данные методы и инструментальные средства не ориентированы на создание агентно-ориентированных систем и не обеспечивают автономного функционирования ИК, хотя и могут быть использованы как средство интеллектуальной интеграции.</p> <p>Методологии проектирования агентно-ориентированных систем до настоящего времени все еще находятся в стадии становления и постановки задач. Проблематике этого направления посвящены работы В.А. Виттиха, В.И. Городецкого, П.О. Скобелева, М. Wooldridge, N.R. Jennings, M. Georgeff, A. Rao, D. Kinny, E.A. Kendall, K. Cetnarowicz, E. Cetnarowicz, E. Nawarecki и др. [194] - [213].</p>
<p>С. 45–46.</p> <p>Відомі підходи можна розділити на три групи: підходи, що базуються на об'єктно-орієнтованих методах і технологіях з використанням відповідних розширень; підходи, які використовують традиційні методи інженерії знань; підходи, засновані на організаційно-орієнтованих уявленнях.</p> <p>У методах першої групи розробляються розширення об'єктно-орієнтованих методів і технологій для проектування агентно-орієнтованих систем. Існує низка CASE-засобів, що підтримують об'єктно-орієнтовані методи розробки ІС, серед яких найбільш відомими є Platinum Paradigm Plus фірми Platinum Technology, а також Rational Rose фірми Rational Software [87–89], процес проектування в яких ґрунтується на мові об'єктно-орієнтованого проектування UML [90–91].</p>	<p>С. 67.</p> <p>Известные подходы можно разделить на три группы:</p> <ul style="list-style-type: none"> - базирующиеся на объектно-ориентированных методах и технологиях с использованием соответствующих расширений; - использующие традиционные методы инженерии знаний; - основанные на организационно-ориентированных представлениях. <p>В методологиях первой группы разрабатываются расширения объектно-ориентированных методологий и технологий для проектирования агентно-ориентированных систем. Существует ряд CASE-средств, поддерживающих объектно-ориентированные методы разработки ИС, среди которых наиболее известными являются PLATINUM Paradigm Plus фирмы PLATINUM technology и Rational Rose фирмы Rational Software [214]-[216], процесс проектирования в которых основывается на языке объектно-ориентированного проектирования UML [217], [218].</p>
<p>С. 46.</p> <p>Відомі спроби безпосереднього застосування UML-нотацій для уявлення агентно-орієнтованих систем і зразків взаємодії агентів [92–93]. Проте ці методи не можуть охопити автономність та активну поведінку агентів, так само як і множини їх взаємодій. З іншого боку, деякі пропонують розширити і адаптувати об'єктно-орієнтовані моделі і технології для визначення методів MAC. Це веде, наприклад, до розширених моделей для</p>	<p>С. 67.</p> <p>Известны попытки непосредственного применения UML-нотации для представления агентно-ориентированных систем и образцов взаимодействия агентов [198], [199].</p> <p>Однако эти предложения не могут охватить автономность и проактивное поведение агентов, так же как и богатство их взаимодействий. С другой стороны, некоторые предлагают расширить и адаптировать объектно-ориентированные модели</p>

<p>представлення агентної поведінки і взаємодії агентів [93–95], так само як агентно-налаштованим розширенням UML [95–96]. Проте, хоча ці пропозиції можуть іноді досягати непоганого моделювання автономної поведінки агентів і їх взаємодій, вони не мають адекватних концептуальних механізмів для роботи з організаціями і співтовариствами агентів.</p>	<p>и технологии для определения методологий MAS. Это ведет, например, к расширенным моделям для представления агентного поведения и взаимодействия агентов [199]-[201], так же как агентно-настроенным расширениям UML [201], [202]. Однако, хотя эти предложения могут иногда достигать хорошего моделирования автономного поведения агентов и их взаимодействий, они не имеют адекватных концептуальных механизмов для работы с организациями и сообществами агентов.</p>
<p>С. 46–47.</p>	<p>С. 67–68.</p>
<p>Велику цікавість представляють інструментальні засоби групи інтелектуальних систем Санкт-Петербурзького інституту інформатики і автоматизації Академії наук Росії [86]. Концептуальною основою що розвивається колективом В.І. Городецького технології розробки MAS є виділення “спільності” функцій, що реалізуються, і подання їх у вигляді класів і структур даних, що є складовою частиною програмного інструментарію. Ідея, що реалізується в цьому підході, полягає у виділенні найбільшої кількості класів і структур даних, повторно використовуваних у агентних застосуваннях. У системі MAS-DK створена конструкція “Типовий агент”, що є базою для подальшої спеціалізації класів і структур даних, клонування екземплярів агентів і індивідуального доопрацювання специфічних компонентів. Цей процес підтримується відповідним набором редакторів.</p>	<p>Среди отечественных разработок большую известность получили инструментальные средства группы интеллектуальных систем Санкт-Петербургского института информатики и автоматизации РАН, возглавляемой В.И. Городецким [196], [197]. Концептуальной основой развиваемой коллективом В.И. Городецкого технологии разработки MAS является выделение «общности» реализуемых функций и представление их в виде классов и структур данных, являющихся составной частью программного инструментария. Реализуемая в этом подходе идея состоит в выделении наибольшего количества классов и структур данных, повторно используемых в агентных приложениях. В системе MASDK создана конструкция «Типовой агент» (“Generic Agent”), являющаяся базой для последующей специализации классов и структур данных, клонирования экземпляров агентов и индивидуальной доработки специфических компонентов. Этот процесс поддерживается соответствующим набором редакторов.</p>
<p>С. 47.</p>	<p>С. 68.</p>
<p>Для управління функціонуванням мультиагентного застосування на всіх етапах життєвого циклу в системі MAS-DK реалізована: концепція “Системного ядра”, що є детальним описом MAS на формальній мові специфікацій з якого можна отримати програмний код в напівавтоматичному режимі. Необхідність в новій генерації програмної коди виникає у випадках модифікації вже існуючої MAS та у ситуаціях аварійного руйнування деяких застосунків. Всі процеси побудови “Системного ядра” засновані на онтології наочної області, що забезпечує несуперечність розподіленої БЗ цільовий MAS.</p> <p>Друга група методів будується на розширенні традиційних методів інженерії знань [97–99]. Ці методи забезпечують формальні і композиційні мови моделювання для верифікації структури системи і функцій. Ці підходи добре застосовні до моделювання знання- і інформаційно-орієнтованих агентів. Проте оскільки ці підходи зазвичай припускають централізований погляд на системи, засновані на знаннях, вони не можуть забезпечити адекватні моделі і підходи для соціального розгляду MAS. Підхід Common Kads намагається перекрити ці обмеження, явно вводючи в методи абстракцію агентного співтовариства. Проте це нововведення зводиться до моделювання співтовариства як колекції взаємодіючої суті без ідентифікації концептів, таких як соціальні завдання або соціальні закони.</p>	<p>Для управления функционированием мультиагентного приложения на всех этапах жизненного цикла в системе MAS DK реализована концепция «Системного ядра», являющегося детальным описанием MAS на формальном языке спецификаций, из которого можно получить программный код в полуавтоматическом режиме. Необходимость в новой генерации программного кода возникает в случаях модификации уже существующей MAS и в ситуациях аварийного разрушения некоторых приложений. Все процессы построения «Системного ядра» основаны на онтологии предметной области, что обеспечивает непротиворечивость распределенной БЗ целевой MAS.</p> <p>Вторая группа методологий строится на расширении традиционных методов инженерии знаний [203]-[205]. Эти методологии обеспечивают формальные и композиционные языки моделирования для верификации структуры системы, и функций. Эти подходы хорошо применимы к моделированию знание- и информационноориентированных агентов. Однако, так как эти подходы обычно предполагают централизованный, взгляд на системы, основанные на знаниях, они не могут обеспечить адекватные модели и подходы для социального рассмотрения MAS. Подход Common Kads пытается перекрыть эти ограничения, явно вводя в методологическую абстракцию агентного сообщества. Однако, это нововведение сводится к моделированию сообщества как коллекции взаимодействующих сущностей без идентификации концептов, таких как социальные задачи или социальные законы.</p>
<p>С. 48.</p>	<p>С. 68–69.</p>
<p>У роботах В.А. Віттіха, П.О. Скобелева, С.В. Ба-</p>	<p>В работах В.А. Виггиха, П.О. Скобелева, С.В.</p>

<p>тіцева, І.А. Мінакова, Г.А. Ржевського запропонована методика створення прикладних відкритих МАС на основі концепції мереж потреб і можливостей (ПМ-мереж) [84–85]. Дана методика має розвинені інструментальні засоби і забезпечує ефективну реалізацію, прикладних МАС в рамках заявленої концепції. Проте вона орієнтована на порівняно вузький клас систем промислової логістики і агентну модель, що описує конкуренцію за ресурси. У ІКСА існує велика кількість типів завдань, що не зводяться до ресурсно-конкурентних моделей [100].</p> <p>Інші моделі і підходи намагаються моделювати і реалізовувати МАС, виходячи з “організаційно-орієнтованої” точки зору [101]. Проте, ці підходи визначають організацію просто як колекцію взаємодіючих ролей, таким чином, не вирішуючи проблеми колективної поведінки агентів.</p>	<p>Батищева, И.А. Минакова, Г.А. Ржевского предложена методика создания прикладных открытых МАС на основе концепции сетей потребностей и возможностей (ПВ-сетей) [194]-[195]1 Данная методика поддержана развитыми инструментальными средствами и обеспечивает эффективную реализацию прикладных МАС в рамках заявленной концепции. Однако, она ориентирована на сравнительно узкий класс систем промышленной логистики, и агентную модель, описывающую конкуренцию за ресурсы. В корпоративных системах ППР существует большое количество типов задач, не сводящихся к ресурсно-конкурентным моделям [31] - [33].</p> <p>Другие модели и подходы пытаются моделировать и реализовывать МАС, исходя из «организационно-ориентированной» точки зрения [207], [208]. Однако, эти подходы определяют организацию просто как коллекцию взаимодействующих ролей, таким образом, не разрешая проблемы коллективного поведения агентов.</p>
<p>С. 48.</p>	<p>С. 71–72.</p>
<p>Достатньо практичним є метод проектування МАС, заснований на концепції М-архітектури, який розвивається в роботах колективу польських: учених на чолі з К. Цетнарівичем [102].</p> <p>У структурі даного методу визначаються такі конструкції як життєвий простір агентів, типи агентів, відносини між агентами і життєвим простором і відноситься з двох різних точок зору як інтелектуальний профіль, що описує здатність агента вирішувати дану проблему і, як енергетичний профіль, що описує, “життєву енергію” агента і що вирішує знищення нежиттєздатних агентів.</p> <p>МАС в даному підході конструюється низхідним способом з використанням абстрактних рівнів, які повинні відповідати наступним правилам:</p> <p>визначення системи задається рівнями специфікації, певними рівнями абстракції, від найбільш загального верхнього рівня до найбільш низького. Нижній абстрактний рівень, який визначає кінцеву версію системи, може бути визначений на мові програмування;</p>	<p>Достаточно оригинальная методология проектирования МАС, основанная на концепции М-архитектуры, развивается в работах коллектива польских ученых во главе с К. Cetnarowicz [211]-[213].</p> <p>В структуре данной методологии определяются такие конструкции как жизненное пространство агентов, типы агентов, отношения между агентами и жизненным пространством и отношения среди агентов. Агентная архитектура рассматривается с двух различных точек зрения: как интеллектуальный профиль, описывающий способность агента решать данную проблему, и как энергетический профиль, описывающий «жизненную энергию» агента и разрешающий уничтожение нежизнеспособных агентов;</p> <p>МАС в рассматриваемом подходе конструируется нисходящим способом с использованием абстрактных уровней, которые должны отвечать следующим правилам:</p> <p>- определение системы задается уровнями спецификации, называемыми уровнями абстракции, от наиболее общего верхнего уровня к наиболее низкому. Нижний абстрактный уровень, который определяет конечную версию системы, может быть определен на языке программирования;</p>
<p>С. 49.</p>	<p>С. 72.</p>
<p>модель ускладнюється і доповнюється новими властивостями у міру просування від верхніх рівнів до нижчих, нові властивості вводяться до розгляду;</p> <p>на даному, абстрактному рівні можуть бути застосовані декілька варіантів абстрактного розгляду, що дозволяє аналізувати дану систему з різних точок зору.</p>	<p>- модель усложняется и дополняется новыми свойствами по мере продвижения от верхних уровней к нижним, новые свойства вводятся в рассмотрение;</p> <p>- на данном абстрактном уровне могут быть применены несколько вариантов абстрактного рассмотрения, что позволяет анализировать данную систему с различных точек зрения.</p>
<p>С. 49.</p>	<p>С. 72.</p>
<p>Автори даного методу постулювали наступні переваги свого підходу: можливість застосування до широкого діапазону МАС від дуже простих до достатньо складних, таких, що включають як простих реактивних так і складних когнітивних агентів; можливість застосування різних математичних формалізмів, можливість створення МАС з кооперацією різних видів агентів (реактивних і когнітивних); незалежність методу від конкретних мов про-</p>	<p>Авторы данной методологии постулируют следующие преимущества своего подхода: возможность применения к широкому диапазону МАС от очень простых до достаточно сложных, содержащих как простых реактивных так и сложных когнитивных агентов; возможность применения различных математических формализмов; возможность создания МАС с кооперацией различных видов агентов (реактивных и когнитивных); независи-</p>

<p>грамування, оскільки середовище програмування може бути визначене на нижньому рівні абстракції разом з програмно-апаратною платформою; гнучкість підходу для аналізу і проектування організацій, що діють, з можливістю декомпозиції завдань на кожному абстрактному рівні, слідуючи організаційним необхідностям.</p> <p>Але, заявлені в роботах [100–102] методи проектування МАС фактично тільки постульовані і висловлюють відомі принципи низхідного проектування, які можуть бути застосовані до дуже широкого спектру систем але не пропонують формалізованих і верифікованих механізмів синтезу.</p>	<p>мость методологии от конкретных языков программирования, так как среда программирования может быть определена на нижнем уровне абстракции вместе с программно-аппаратной платформой; гибкость подхода для анализа и проектирования действующих организаций, с возможностью декомпозиции задач на каждом абстрактном уровне, следуя организационным потребностям.</p> <p>По мнению автора, заявленная в работах [211]-[213] методология проектирования МАС фактически только постулирована и излагает известные принципы нисходящего проектирования, действительно применимые к очень широкому спектру систем и не предлагает формализованных и верифицированных механизмов синтеза.</p>
<p>С. 49–50.</p>	<p>С. 73.</p>
<p>1.4. Постановка завдань дисертаційного дослідження</p> <p>Відомі підходи і методи інформаційної взаємодії в ІС не забезпечують підвищення інтелектуального рівня ІКСА. Не дивлячись на низку важливих робіт в області побудови інтегрованих інтелектуальних систем В.Ф. Горнева, А.В. Колесникова, Е.В. Попова, Г.В. Рыбиной, В.Ф. Хорошевського, А.И. Эрлиха, завдання створення методики проектування інтелектуальних інформаційно-керуючих систем для таких складних об'єктів, як аеропорт, повністю не вирішено. Вимагають подальшого дослідження і розробки такі питання як:</p> <ul style="list-style-type: none"> - створення методики побудови ІКСА, що охоплює всі етапи аналізу, проектування і реалізації багатокomпонентних додатків; - реалізації мовних і інструментальних засобів (універсальних і проблемно-орієнтованих), що підтримують повний життєвий цикл розробки ІКСА; - розробка узагальнюючих уявлень для різних класів моделей і методів розробки ІКСА на цих моделях; - створення методів і процедур автоматизованого придбання знань з урахуванням особливостей ІКСА. 	<p>1.5. Формулировка задач диссертационного исследования</p> <p>Проблема исследования и разработки моделей и методов интеллектуальной интеграции в КСППР в настоящее время находится в стадии формирования концептуальных установок и постановки задач.</p> <p>Известные подходы и методы информационной интеграции в КИС не обеспечивают повышение интеллектуального уровня КСППР. Несмотря на ряд важных работ в области построения интегрированных интеллектуальных систем В.Ф. Горнева, А.В. Колесникова, Э.В. Попова, Г.В. Рыбиной, В.Ф. Хорошевского, А.И. Эрлиха, проблема создания общей методологии проектирования данного класса систем полностью не решена. Требуют дальнейшего исследования и разработки такие вопросы как:</p> <ul style="list-style-type: none"> - создание методологии разработки ИИС, охватывающей все этапы анализа, проектирования и реализации многокомпонентных приложений; - реализации языковых и инструментальных средств (универсальных и проблемно-ориентированных), поддерживающих полный жизненный цикл разработки ИИС; - разработка обобщающих представлений для различных классов моделей и методов ППР на этих моделях; - создание методов и процедур автоматизированного приобретения знаний с учетом особенностей ИИС;
<p>С. 50.</p>	<p>С. 73.</p>
<p>Для формалізації завдання інтелектуальної взаємодії в ІКСА введемо наступні позначення:</p> $AS = \{ASSm\}, (1.8)$ <p>де AS – ІКСА, ASSm – m-а підсистема AS, що характеризуються функціональними відмінностями (управлінські, організаційно-економічні, технологічні, технічні, екологічні і інші функціональні підсистеми та служби аеропорту). У свою чергу, ASSm = {ASSm,p}, де ASSm,p – підсистеми у складі ASSm, які можуть розглядатися ІКС, як системи автоматизованого проектування, системи управління базами даних і різні застосування.</p>	<p>Для формализации задачи интеллектуальной интеграции КИНС ППР введем следующие обозначения:</p> $CS = \{CSSm\}, (1.3)$ <p>где CS - корпоративная система, CSSm - m-я подсистема CS, характеризующиеся функциональными различиями (организационно-экономические, технологические, научно-исследовательские, экологические и другие функциональные подсистемы).. В свою очередь, CSSm = {CSSm,p}, где CSSm,p - подсистемы в составе CSSm, в качестве которых могут рассматриваться СППР, САПР, СУБД и различные приложения.</p>
<p>С. 51.</p>	<p>С. 74.</p>
<p>Нехай $I_{CS} \left(i, \bigcup_{l \in L_1} K_l \right)$ – модель (проект) ІНС на і-му</p>	<p>Пусть $I_{CS}(i, \bigcup_{l \in L_1} K_l)$ - модель (проект) КИНС ППР на i-том этапе построения в терминах используе-</p>

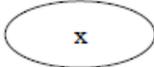
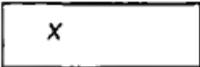
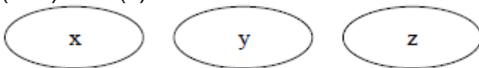
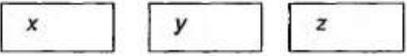
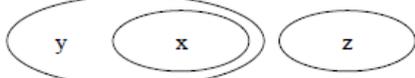
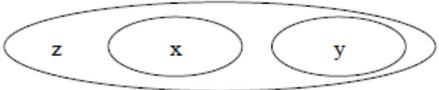
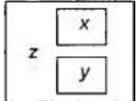
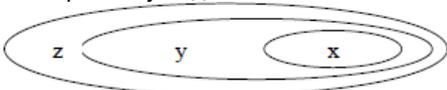
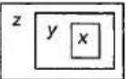
<p>етапі побудови в термінах використовуваних мов $K_i, i = 1, \dots, L_i; i = 1, \dots, E$, де E – число етапів побудови. Процес побудови представимо у вигляді</p> $I_{CS} \left(i, \bigcup_{l \in L_i} K_l \right) \Rightarrow \Pi(i, i+1) \Rightarrow I_{CS} \left(i+1, \bigcup_{l \in L_{i+1}} K_l \right) \quad (1.10)$ <p>де $\Pi(i, i+1)$ – оператор побудови, що визначає створення моделі (проекту) системи в термінах відповідних мов ($K = \{K_i\}; K = \{\Pi(i, i+1)\}$).</p> <p>У загальному випадку завдання побудови ІКСА є складним і багатоаспектним. Для таких систем досить важко побудувати обґрунтований функціонал корисності, тому необхідні умови досягнення властивостей взаємодії (властивостей цілісності) ІНС необхідно сформулювати при певних обмеженнях.</p>	<p>мых языков $K_i, i = 1, \dots, L_i = 1, \dots, E$, где E - число этапов построения.</p> <p>Процесс построения представим в виде</p> $I_{CS} \left(i, \bigcup_{l \in L_i} K_l \right) \xrightarrow{\Pi(i, i+1)} I_{CS} \left(i+1, \bigcup_{l \in L_{i+1}} K_l \right), \quad (1.4)$ <p>где $\Pi(i, i+1)$ - оператор построения, определяющий создание модели (проекта) системы в терминах соответствующих языков ($K = \{K_i\}, \Pi = \{\Pi(i, i+1)\}$).</p> <p>В общем случае проблема построения КИНС ППР является сложной и многоаспектной. Для таких систем весьма трудно построить обоснованный функционал полезности, поэтому автор формулирует необходимые условия достижения интегративных свойств (свойств целостности) КИНС при определенных ограничениях.</p>
<p>С. 51–52.</p>	<p>С. 74–75.</p>
<p>У даній роботі пропонується вирішення завдання щодо забезпечення інтелектуальної взаємодії в ІКСА на основі агентноорієнтованого підходу спільно з методами розподіленого штучного інтелекту і інженерії знань.</p> <p>Нечисленні відомі методи побудови агентноорієнтованих систем, викладені в роботах В.А. Віттіха, В.І. Городецького, П.О. Скобелева, М. Вулдріджа, Н.Р. Дженінгса, М. Георгієва, А. Рао, Д. Кіні, Е.А. Кендала, К. Цетнарівича, Е. Наварескі, А.Н. Швецова не можуть охопити весь комплекс завдань побудови ІНС, оскільки засновані на загальних концепціях і математичних моделях та не враховують ієрархічний характер інтелектуальної взаємодії в ІКСА.</p>	<p>В настоящей работе автор предлагает решение проблемы интеллектуальной интеграции в КС на основе концепции КИНС ППР, использующей агентно-ориентированный подход совместно с методами распределенного искусственного интеллекта и инженерии знаний.</p> <p>Немногочисленные известные методологии построения агентно-ориентированных систем, изложенные в работах В.А. Виттиха, В.И. Городецкого, И.О. Скобелева, М. Wooldridge, N.R. Jennings, M. Georgeff, A. Rao, D. Kinny, E.A. Kendall, K. Sctnarowicz, E. Nawarecki не могут охватить весь комплекс задач построения КИНС ППР, так как основаны на других концепциях и математических моделях, и не учитывают иерархический характер интеллектуальной интеграции в КС.</p>
<p>С. 52.</p>	<p>С. 75.</p>
<p>У формальній постановці завдання інтелектуальної взаємодії і підвищення інтелектуального рівня в ІКС визначається таким чином. Початкові дані:</p> <p>визначена ІКСА відповідно до формул (1.1 – 1.4, 1.8);</p> <p>ІКСА володіє властивістю інформаційної взаємодії по (1.5);</p> <p>визначена множина ОПР, для яких необхідна підтримка ІКС і середня кількість завдань прийняття рішення на одну ОПР за час $T_c - TR$;</p> <p>встановлений потенційний запас працездатності ІКС $Z = \{(M_{ch}(i) - M_d(i)) i = 1, \dots, N\}$, де $M_d(i)$ – середнє необхідне число завдань прийняття рішень на одну ОПР, N – число ОПР в ІКС.</p>	<p>В формальной постановке проблема интеллектуальной интеграции и повышения интеллектуального уровня в КСППР определяется следующим образом.</p> <p>Исходные данные:</p> <ul style="list-style-type: none"> - определена КС в соответствии с формулой (1.3); - известно, что она обладает свойством информационной интеграции по (1.1); - определено множество ЛПР, для которых необходима поддержка СППР и среднее количество ЗПР на одно ЛПР за время $T_C - TR$; - установлен потенциальный запас работоспособности КСППР - $Z = \{(M_{ch}(i) - M_d(i)) i = 1, \dots, N\}$, где $M_d(i)$ - среднее требуемое число ЗПР на одно ЛПР, N - число ЛПР в КСППР.
<p>С. 52.</p>	<p>С. 75.</p>
<p>Необхідно визначити процес побудови ІКСА (1.10), який веде до побудови моделі ІНС що володіє властивістю інтелектуальної взаємодії і забезпечує потенційний запас працездатності для всіх ОПР в ІКС.</p> <p>Стратегічний план побудови такої ІКСА полягає в розробці відповідної структурно-логічної моделі, в якій кожній ОПР буде зіставлена необхідна кількість ІА, що приймають рішення з визначеного переліку завдань, з тим щоб забезпечити потенційний запас працездатності. Інтелектуальні агенти в ІКСА повинні володіти рівнем інтелекту достатнім для підтримки інтелектуальної взаємодії в ІКС.</p>	<p>Необходимо определить процесс построения КИНС ППР в соответствии с (1.4), ведущий к построению модели КИНС ППР, обладающей свойством интеллектуальной интеграции и обеспечивающей потенциальный запас работоспособности для всех ЛПР в КС.</p> <p>Стратегический план построения такой КИНС ППР состоит в разработке соответствующей структурно-логической модели, в которой каждому ЛПР будет сопоставлено необходимое количество ИА, принимающих на себя решение ряда ЗПР, с тем чтобы обеспечить потенциальный запас работоспособности. ИА в КИНС ППР должны обладать уровнем интеллекта достаточным для поддержа-</p>

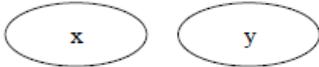
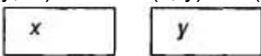
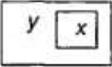
		ния интеллектуальной интеграции в КС.
С. 52–53.	Для вирішення поставленого завдання у дисертаційній роботі необхідно дослідити і розробити: типи основних інтелектуальних компонентів ІІКСА; математичні моделі інтелектуальних компонентів ІІКСА; методику побудови ІІКСА; базову архітектуру ІІКСА і алгоритми логічного виводу в даній архітектурі; рекомендації щодо побудови інструментального комплексу для побудови ІІКСА.	С. 75–76. Для решения поставленной проблемы автор диссертационной работы исследует и разрабатывает: - концепцию корпоративной интеллектуальной системы поддержки принятия решений, устанавливает уровни интеллектуальной иерархии и обобщает типы интеллектуальных компонентов КИНС ППР; - математические модели интеллектуальных компонентов КИНС ППР; - методологию построения КИНС ППР; - базовую архитектуру корпоративной интеллектуальной системы для поддержки принятия решений и алгоритмы логического вывода в данной архитектуре; - инструментальный комплекс для построения КИНС ППР.
С. 53–54.	Висновки до розділу 1 <...> Разом з тим впровадження існуючих засобів штучного інтелекту не є тривіальним завданням. Основною вимогою щодо їх впровадження є необхідність забезпечення інтелектуальної взаємодії в інформаційно-керувальній системі аеропорту. На підставі проведеного аналізу запропоновано точне формальне визначення даного поняття. Вивчення і дослідження розподіленої об'єктної архітектури дозволяє стверджувати, що така архітектура забезпечує необхідний програмно-апаратний базис для інтелектуальної взаємодії, підтримуючи на рівні додатків моделі мережевої взаємодії інформаційної системи, дозволяючи надбудувати над ними рівень інтелектуальної взаємодії розподілених компонентів.	С. 76. Выводы по главе 1 <...> Выявлена и проанализирована проблема интеллектуальной интеграции в корпоративных СППР. На основании проведенного анализа автором предложено точное формальное определение данного понятия. Изучение и исследование распределенных объектных архитектур позволяет утверждать, что такие архитектуры обеспечивают необходимый программно-аппаратный базис для интеллектуальной интеграции, поддерживая уровни представлений и приложений модели сетевого взаимодействия OSI, позволяя надстраивать над ними уровни интеллектуальной интеграции распределенных компонентов.
С. 54.	(Висновки до розділу 1) Побудова інтелектуальної інформаційно-керувальної системи аеропорту може бути здійснена на базі мультиагентної архітектури, яка об'єднує переваги двох характерних напрямків побудови інформаційних систем: розподіл інтелектуальних функцій по окремих компонентах системи (розподілений штучний інтелект, паралельний штучний інтелект) і взаємодії, об'єднання різних видів інформаційної системи в єдиному інформаційному просторі авіа-підприємства (інтегровані, гібридні інформаційні системи).	С. 77. (Выводы по главе 1) В развитии современных ИИС автор диссертационной работы прослеживает две тенденции: распределения интеллектуальных функций по отдельным компонентам ИИС (распределенный искусственный интеллект, метакомпьютер, параллельный искусственный интеллект) и интеграции, объединения различных видов ИИС в едином информационном пространстве корпорации, предприятия, организации, (интегрированные ИИС, гибридные ИИС).
С. 54.	(Висновки до розділу 1) Ідея побудови інтелектуальної інформаційно-керувальної системи аеропорту базується на сумісному використанні наступних підходів: агентної орієнтації, що полягає у переважному використанні як основного компонента множини інтелектуальних агентів, які володіють автономним функціонуванням і цілеспрямованою поведінкою; ієрархічної інтелектуальної організації компонентів інформаційно-керувальної системи, яка припус-	С. 77. (Выводы по главе 1) В разделе 1.3. разработана концепция корпоративной интеллектуальной системы поддержки принятия решений, базирующаяся на совместном использовании следующих системообразующих принципов: - принципа агентной ориентации, заключающегося в преимущественном использовании в качестве основных компонентов СППР интеллектуальных агентов, обладающих автономным функционированием и целенаправленным поведением; - принципа иерархической интеллектуальной

<p>кає наявність в системі агентів різної інтелектуальної ваги і підпорядкування агентів з меншою вагою агентам з більшою вагою;</p> <p>мультиплікативності інтелектуальних агентів в інформаційно-керуючій системі аеропорту, яка припускає створення для кожної посадової особи авіа-підприємства такої кількості агентів, яка необхідна для того, щоб забезпечити необхідний запас продуктивності.</p>	<p>организации компонентов СППР, предполагающего наличие в системе агентов различного интеллектуального веса и подчинение ИА с меньшим весом агентам с большим весом;</p> <p>- принципа мультипликативности интеллектуальных агентов в СППР, предполагающего создание для каждого ЛПР такого количества ИА, которое необходимо для того, чтобы обеспечить требуемый запас производительности.</p>
<p>С. 54–55.</p>	<p>С. 78.</p>
<p>(Висновки до розділу 1)</p> <p>Для формування методики синтезу інтелектуальної інформаційно-керуючої системи аеропорту необхідно розробити: логічну структуру, що відображає організацію інтелектуальних компонентів і їх взаємодію з суб'єктами авіапідприємства і об'єктами інформаційної технології; об'єктну структуру, що визначає організацію розподілених об'єктів в системі управління; топологічну структуру, що визначає просторові взаємозв'язки інтелектуальних компонентів і об'єктів в певній мережевій архітектурі. Також необхідно сформулювати завдання подальшого дослідження, що полягає в дослідженні і розробці моделей інтелектуальних складових інформаційно-керуючої системи, запропонованих в її концептуальній структурі.</p>	<p>(Выводы по главе 1)</p> <p>Разрабатываются представления о необходимых структурных репрезентациях КИНС ППР: логической структуре, отображающей организацию интеллектуальных компонентов и их взаимодействие с субъектами и объектами информационной технологии; объектной структуре, определяющей организацию распределенных объектов в системе; топологической структуре, определяющей пространственные взаимосвязи интеллектуальных компонентов и объектов в определенной сетевой архитектуре. Формулируются задачи последующего исследования, заключающиеся в исследовании и разработке моделей интеллектуальных составляющих КИНС ППР, постулированных в концептуальной структуре КИНС ППР.</p>
<p>С. 55.</p>	<p>С. 78.</p>
<p>(Висновки до розділу 1)</p> <p>Внаслідок того, що інтелектуальна інформаційно-керуюча система аеропорту є новим видом інформаційних систем необхідним є дослідження моделей і архітектури баз знань, організація яких відрізнятиметься від відомих інтелектуальних систем. Також необхідно досліджувати методи логічного виводу в конфігурації розподілених баз знань інтелектуальної системи аеропорту, оскільки моделі логічного виводу відрізнятимуться від традиційних моделей, вживаних в інформаційних системах інших областей застосування з одного боку, і від моделей, вживаних в репрезентаціях програмних агентів, з іншого боку.</p> <p>Виконаний в першому розділі аналіз відомих методів інформаційної взаємодії і найбільш розроблених методів проектування мультиагентних систем показує, що існуючі підходи не в змозі вирішити весь комплекс завдань побудови інтелектуальної інформаційно-керуючої системи аеропорту, оскільки відрізняються в концептуальних установках, цілях і використовуваних математичних моделях.</p> <p>Тому актуальним є наукове завдання щодо створення методики побудови інтелектуальної інформаційно-керуючої системи аеропорту, заснованої на методах штучного інтелекту та інженерії знань, яка враховує досягнення і переваги проаналізованих вище підходів, відповідає заявленим концептуальним установкам і дозволяє автоматизувати основні етапи процесу розробки.</p>	<p>(Выводы по главе 1)</p> <p>Вследствие того, что КИНС ППР является новым видом ИНС необходимо исследование моделей и архитектуры баз знаний, организация которых будет отличаться от известных интеллектуальных систем. Также необходимо исследовать методы логического вывода в конфигурации распределенных баз знаний КИНС ППР, так как модели логического вывода будут отличаться от традиционных моделей, применяемых в ЭС с одной стороны, и от моделей, применяемых в интенциональных репрезентациях программных агентов, с другой стороны.</p> <p>Выполненный в первой главе анализ известных методов информационной интеграции, изложенных в стандартах IDEF, и наиболее разработанных методологий проектирования МАС показывает, что существующие подходы, рассматриваемые по отдельности, не в состоянии решить весь комплекс задач построения КИНС ППР, поскольку отличаются в концептуальных установках, преследуемых целях и используемых математических моделях.</p> <p>Поэтому автор диссертационной работы ставит перед собой задачу создания методологии построения КИНС ППР, учитывающей достижения и достоинства обозначенных выше подходов, соответствующей заявленным концептуальным установкам и позволяющей автоматизировать основные этапы процесса разработки.</p>
<p>С. 56.</p>	<p>С. 83.</p>
<p>У якості методу побудови програмних продуктів для інформаційно-керуючої системи аеропорту обрано об'єктно-орієнтоване програмування. Об'єктно-орієнтована термінологія визначає об'єкт як пакет, що містить дані, процедури та методи, які надаються як сервіси іншим об'єктам. Абстракція</p>	<p>Традиционная объектно-ориентированная терминология определяет объект как пакет, содержащий данные и процедуры, имеющий методы, которые предоставляются как сервисы другим объектам. Абстракция объектов во многом направлена</p>

	на эффективное выполнение распределенных вычислений, которые в объектно-ориентированной системе тесно связаны с обменом сообщениями между объектами.
С. 56.	С. 87.
<p>Побудова формальної об'єктної системи. Необхідним базисом для формалізації інтелектуальних компонентів ІКСА є модель інформаційного об'єкта (ІО), яка повинна будуватися в рамках деякої формальної системи (теорії). Як математичний апарат такої теорії (числення об'єктів) доцільно використати логіку першого порядку.</p> <p>Нехай α – множина функціональних, предикативних і константних символів. Кожному, функціональному символу $f \in \alpha$ можна поставити у відповідність ціле позитивне число $\#(f)$ таке, що якщо $n = \#(f)$, то f називатимемо n-арним функціональним символом. Кожен предикативний символ $R \in \alpha$ можна пов'язати з позитивним цілим числом $\#(R)$; якщо $n = \#(R)$, то R називатимемо n-арним предикативним символом. Предикативні символи за своєю суттю є символами відносин. Відповідно до [105], під алгебраїчною системою Φ розумітимемо непорожню сукупність M об'єктів, яка є областю дії кванторів, разом з інтерпретацією основних предикативних, функціональних і константних символів з α.</p>	<p>2.2.2. Построение формальной объектной аксиоматизируемой системы</p> <p>Таким образом, необходимым базисом для формализации КИНС ППР является модель ИО, которая должна строиться в рамках формальной аксиоматизируемой системы (теории), основания которой и рассматриваются далее. В качестве математического аппарата такой теории (исчисления объектов) автор использует логику первого порядка.</p> <p>Пусть α - множество функциональных, предикатных и константных символов, конечное или счетно-бесконечное (что оставляет возможность расширения языка α). Каждому функциональному символу $f \in \alpha$ можно поставить в соответствие целое положительное число $\#(f)$ такое, что если $n = \#(f)$, то f называется n-арным функциональным символом. Каждый предикатный символ $R \in \alpha$ можно связать с положительным целым числом $\#(R)$; если $n = \#(R)$, то R называется n-арным предикатным символом. Предикатные символы по своей сути являются символами отношений. В соответствии с [258], под алгебраической системой Φ будем понимать непустую совокупность M объектов, которая является областью действия кванторов, вместе с интерпретацией основных предикатных, функциональных и константных символов из α.</p>
С. 57.	С. 87.
<p>Алгебраїчна система для мови α є пара $\Phi = \langle M, F \rangle$, де M непорожня множина, F – відображення з області визначення α таке, що:</p> <ol style="list-style-type: none"> якщо $R \in \alpha$ – n-арний предикативний символ, то $F(R) \subseteq M^n$; якщо $f \in \alpha$ – n-арний функціональний символ, то $F(f): M^n \rightarrow M$; якщо $c \in \alpha$ – константний символ, то $F(c) \in M$. <p>Основними синтаксичними поняттями логіки 1-го порядку є: логічні зв'язки $\&, \vee, \neg, \rightarrow, \leftrightarrow$; квантори спільності і існування \forall і \exists, символи логічних змінних x, y, z, \dots. Всяку скінчену послідовність, елементами якої є основні символи або елементи, будемо назвати виразом.</p>	<p>Определение 1. Алгебраическая система для языка α есть пара $\Phi = \langle M, F \rangle$, где M - непустое множество, F - отображение с областью определения α такое, что:</p> <ol style="list-style-type: none"> если $R \in \alpha$ - n-арный предикатный символ, то $F(R) \subseteq M^n$; если $f \in \alpha$ — n-арный функциональный символ, то $F(f): M^n \rightarrow M$; если $c \in \alpha$ - константный символ, то $F(c) \in M$. <p>Основными синтаксическими понятиями логики 1-го порядка являются: логические связи $\&, \vee, \neg, \rightarrow, \leftrightarrow$; кванторы общности и существования \forall и \exists, символы логических переменных, обозначаемых x, y, z, \dots. Всякую конечную последовательность, элементами которой являются основные символы или элементы α, можно назвать выражением.</p>
С. 57.	С. 87–88.
<p>Терми мови α утворюють найменшу множину виразів, що містять x, y, z, всі константні символи α (якщо такі є) і замкнуту відносно правила утворення: якщо t_1, \dots, t_n – терми α і якщо $f \in \alpha$ – n-місний функціональний символ, то вираз $f(t_1, \dots, t_n)$ є термом мови α. Терм, що не містить змінних, називатимемо замкнутим.</p> <p>Атомарна формула мови α – це вираз одного з наступних видів:</p> $(t_1 = t_2), R(t_1, \dots, t_n),$ <p>де t_1, t_2 – терми мови, а $R \in \alpha$ — довільний n-місний предикативний символ.</p>	<p>Определение 2. Термы языка α образуют наименьшее множество выражений, содержащее x, y, z, \dots, все константные символы α (если таковые имеются) и замкнутое относительно правила образования: если t_1, \dots, t_n - термы α и если $f \in \alpha$ - n-местный функциональный символ, то выражение $f(t_1, \dots, t_n)$ является термом языка α. Терм, не содержащий переменных, называется замкнутым.</p> <p>Определение 3. Атомная формула языка α - это выражение одного из следующих видов:</p> $(t_1 = t_2), R(t_1, \dots, t_n),$ <p>где t_1, t_2 - термы языка α, а $R \in \alpha$ - произвольный n-местный предикатный символ.</p>
С. 57.	С. 88.

	<p>Формули першого порядку мови α утворюють найменшу множину виразів, що містить атомарні формули і замкнуту щодо наступного правила утворення:</p> <p>1) якщо ϕ і ψ – формули, то вирази $\neg\phi$, $(\phi \& \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$ також є формулами.</p> <p>2) якщо ϕ – формула і v – змінна, то $(\exists v\phi)$ і $(\forall v\phi)$ також є формулами.</p> <p>Множина $Fv(\phi)$ вільних змінних формули ϕ визначається таким чином:</p>	<p>Определение 4. Формулы первого порядка языка α образуют наименьшее множество выражений, содержащее атомные формулы и замкнутое относительно следующего правила образования:</p> <p>1) если ϕ и ψ – формулы, то выражения $\neg\phi$, $(\phi \& \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$ также являются формулами.</p> <p>2) если ϕ – формула и v – переменная, то $(\exists v\phi)$ и $(\forall v\phi)$ также являются формулами.</p> <p>Определение 5. Множество $Fv(\phi)$ свободных переменных формулы ϕ определяется следующим образом:</p>
С. 58.		С. 88.
	<p>1) якщо ϕ – атомарна формула, то $Fv(\phi)$ в точності множина змінних, що зустрічаються в ϕ;</p> <p>2) $Fv(\neg\phi) = Fv(\phi)$;</p> <p>3) $Fv(\phi \& \psi) = Fv(\phi) \cup Fv(\psi)$;</p> <p>4) $Fv(\exists v\phi) = Fv(\forall v\phi) = Fv(\phi) - \{v\}$.</p> <p>Реченням (першого порядку) мови α називаємо формулу, що не містить вільних змінних.</p> <p>Будуємо систему алгебри для мови логіки 1-го порядку α над множиною M-інформаційних об'єктів ІКСА. Змінними позначаються ІО, далі вводяться предикативні і функціональні символи над множиною M.</p>	<p>1) если ϕ – атомная формула, то $Fv(\phi)$ в точности множество переменных, встречающихся в ϕ;</p> <p>2) $Fv(\neg\phi) = Fv(\phi)$;</p> <p>3) $Fv(\phi \& \psi) = Fv(\phi \vee \psi) = Fv(\phi) \cup Fv(\psi)$;</p> <p>4) $Fv(\exists v\phi) = Fv(\forall v\phi) = Fv(\phi) - \{v\}$.</p> <p>Определение 6. Предложением (первого порядка) языка α называется формула, не содержащая свободных переменных.</p> <p>Строим алгебраическую систему для языка логики 1-го порядка α над множеством M – информационных объектов. Переменными обозначаются ІО, далее вводятся предикатные и функциональные символы над множеством M.</p>
С. 58.		С. 88–89.
	<p>Інформаційний об'єкт пропонується визначити наступним чином:</p> <p>$O := \langle NO, \{A\}, \{O\}, \{F\} \rangle$</p> <p>де NO – ім'я об'єкта; $\{A\}$ – множина атрибутів об'єкта (A_0, \dots, A_n), де A_i – i-й атрибут ІО; $\{O\}$ – множина об'єктів, які структурно входять до даного об'єкта, $(O_{NO1}, O_{NO2}, O_{NOm})$, де O_{NOi} – i-й підпорядкований об'єкт, об'єкту з ім'ям NO; $\{F\}$ – множина функцій, які виконує даний ІО.</p>	<p>Информационный объект определяется следующим образом:</p> <p>$O := \langle \text{имя_объекта}, \{A\}, \{O\}, \text{модель_поведения} \rangle$.</p> <p>Введем более компактные обозначения: имя объекта (Name of Object) = NO, модель поведения (Behavior Model) = BM и получим:</p> <p>$O := \langle NO, \{A\}, \{O\}, BM \rangle$,</p> <p>где NO – символьная строка, соответствующая соглашению об именах, вводимому далее; $\{A\}$ – множество атрибутов объекта (A_0, A_1, \dots, A_n), где A_i – i-й атрибут ІО; $\{O\}$ – множество объектов, вложенных в данный объект (в смысле структурного вложения) $(O_{NO1}, O_{NO2}, \dots, O_{NOm})$, где O_{NOi} – i-вложенный объект, объекта с именем NO.</p>
С. 58–59.		С. 89.
	<p>Атрибут ІО визначимо як:</p> <p>$A = \langle NA, SA, VA \rangle$</p> <p>де NA – ім'я атрибута, SA – множина, на якій визначається значення атрибута, VA – значення атрибута, тобто $a \in SA$ в даний момент часу t.</p> <p>Примітивним ІО будемо називати такий ІО, що $O = \langle NO, \{A\}, \emptyset, \{F\} \rangle$, тобто що має порожню множину вкладених ІО. Можна ввести структурний ІО (або пасивний), який визначається як $O = \langle NO, \{A\}, \emptyset, \emptyset \rangle$, тобто який має порожню множину вкладених ІО і порожню множину функцій, а також ввести однопараметричний ІО: $O = \langle NO, A, \emptyset, \emptyset \rangle$ для скорочення запису що позначається як $\langle NO, A \rangle$, що має тільки один атрибут. Таке визначення ІО дозволяє розглядати множину M алгебраїчної системи як актуальну і залишатися в рамках логіки 1-го порядку при розгляді відносин між ІО.</p>	<p>Атрибут ІО определим как:</p> <p>$A = \langle NA, SA, VA \rangle$,</p> <p>где NA – имя атрибута (символьная строка, соответствующая соглашению об именах), SA – множество, на котором определяется значение атрибута, VA – значение атрибута, т. е. $a \in SA$ в данный момент времени t.</p> <p>Примитивным ІО можно назвать такой ІО, что $O = \langle NO, \{A\}, \emptyset, \emptyset \rangle$, т. е. имеющий пустое множество вложенных ІО. Можно ввести структурный ІО (или пассивный), определяемый как $O = \langle NO, \{A\}, \emptyset, \emptyset \rangle$, т. е. имеющий пустое множество вложенных ІО и пустую модель поведения, а также ввести однопараметрический ІО: $O = \langle NO, A, \emptyset, \emptyset \rangle$, для сокращения записи обозначаемый как $\langle NO, A \rangle$, имеющий только один атрибут. Такое определение ІО позволяет рассматривать множество M алгебраической системы как актуальное и оставаться в рамках логики 1-го порядка при рассмотрении отношений между ІО.</p>
С. 59.		С. 89.
	<p>Почнемо введення відносин в дану модель Φ з відношення належності об'єктів (інакше кажучи,</p>	<p>Начнем введение отношений в рассматриваемую модель Φ с отношения принадлежности объ-</p>

<p>відношення структурної підпорядкованості) Rs.</p> <p>Аксиома R.1: Відношення Rs – антирефлексивне: $\Box x(\Box Rs(x, x))$. Це витикає із змістовного поняття структурної вкладеності (сам предмет не містить самого себе у якості структурного елемента).</p> <p>Аксиома R.2: Відношення Rs – транзитивне: $\Box x\Box y\Box z(Rs(x, y)\&Rs(y, z)\rightarrow Rs(x, z))$.</p> <p>Аксиома R.3: Відношення Rs – несиметричне: $\Box x\Box y(Rs(x, y)\rightarrow \Box Rs(y, x))$.</p> <p>Розглядаючи ці властивості як аксиоми в численні предикатів, покажемо, що вони є загальнозначущими формулами в мові α (або тавтологіями).</p> <p>Для аксиоми R.1: $\Box x(\Box Rs(x, x))$. Скористаємося геометричною інтерпретацією:</p> 	<p>ектов (інаше говоря, отношение структурной вложенности) Rs. Отношение Rs - антирефлексивно: $\forall x(\neg Rs(x, x))$. Это следует из содержательного понятия структурной вложенности (сам предмет не содержит самого себя как структурной части). Отношение Rs - транзитивно и несимметрично:</p> <p>$\forall x\forall y\forall z(Rs(x, y)\&Rs(y, z) \rightarrow Rs(x, z))$.</p> <p>$\forall x\forall y(Rs(x, y) \rightarrow \neg Rs(y, x))$.</p> <p>Рассматривая эти свойства как аксиомы в численнии предикатов, покажем, что они являются общезначимыми формулами в языке α (или тавтологиями).</p> <p>Для аксиомы Rs(1): $\forall x\neg Rs(x, x)$. Воспользуемся очевидной геометрической интерпретацией:</p> 
<p>С. 59–60.</p>	<p>С. 90.</p>
<p>Для самого себе об'єкт x не забезпечує істинності Rs. Тому для всіх x аксіома Rs(1) є тавтологією.</p> <p>Для аксіоми R.2 розглянемо наступні варіанти структурних відносин:</p> <p>Варіант 1. Об'єкти x, y і z незалежні один від одного. Формулу $(Rs(x, y)\&Rs(y, z)\rightarrow Rs(x, z))$ розкриваємо як диз'юнкцію відповідно до правил числення предикатів, тоді отримуємо:</p> <p>$(\neg(Rs(x, y)\&Rs(y, z))\vee Rs(x, z))$. При даних відносинах об'єктів маємо істинні оцінки $R_s(x, y) = f$, $R_s(y, z) = f$, $R_s(x, z) = f$ отже $\neg(f \& f) \vee f = \neg(f \& f) \vee f = t \vee f = t$.</p> 	<p>Для самого себя объект x не обеспечивает истинности Rs. Поэтому для всех x аксиома Rs(1) является тавтологией.</p> <p>Для аксиомы Rs(2) рассмотрим следующие варианты структурных отношений:</p>  <p>Вариант 1. Объекты x, y и z независимы друг от друга. Формулу $(Rs(x, y)\&Rs(y, z)\rightarrow Rs(x, z))$ раскрываем как диз'юнкцию в соответствии с правилами ИП, тогда получаем: $(\neg(Rs(x, y)\&Rs(y, z))\vee Rs(x, z))$. При данных отношениях объектов, имеем истинные оценки $R_s(x, y) = f$, $R_s(y, z) = f$, $R_s(x, z) = f$; следовательно, $\neg(f \& f) \vee f = \neg(f \& f) \vee f = t \vee f = t$.</p>
<p>С. 60.</p>	<p>С. 90.</p>
<p>Варіант 2. Об'єкт y включає x, а z – незалежний. Тоді $R_s(x, y) = t$, $R_s(y, z) = f$, $R_s(x, z) = f$, і отже $\neg(t \& f) \vee f = \neg(t \& f) \vee f = t \vee f = t$. З точністю до перейменування змінних, аналогічні результати виходять для випадків включення одного об'єкта в інший і незалежності третього.</p> 	 <p>Вариант 2. Объект y включает x, а z - независим. Тогда $R_s(x, y) = t$, $R_s(y, z) = f$, $R_s(x, z) = f$ и следовательно, $\neg(t \& f) \vee f = \neg(t \& f) \vee f = t \vee f = t$. С точностью до переименования переменных, аналогичные результаты получаются для случаев включения одного объекта в другой и независимости третьего.</p>
<p>С. 60.</p>	<p>С. 90.</p>
<p>Варіант 3. Два об'єкти x і y входять в об'єкт z. В цьому випадку $R_s(x, y) = f$, $R_s(y, z) = t$, $R_s(x, z) = t$ $\neg(f \& t) \vee t = \neg(f \& t) \vee t = t \vee t = t$. З точністю до перейменування змінних, отримуємо докази істинності формул, аналогічно варіанту 2.</p> 	<p>Вариант 3. Два объекта x и y входят в объект z. В этом случае $R_s(x, y) = f$, $R_s(y, z) = t$, $R_s(x, z) = t$, $\neg(f \& t) \vee t = \neg(f \& t) \vee t = t \vee t = t$. С точностью до переименования переменных, получим доказательства истинности формул, аналогично варианту 2.</p> 
<p>С. 60.</p>	<p>С. 90.</p>
<p>Варіант 4. Об'єкт x входить в об'єкт y, а y входить в z. Маємо $R_s(x, y) = t$, $R_s(y, z) = t$, $R_s(x, z) = t$, тоді $\neg(t \& t) \vee t = \neg(t \& t) \vee t = f \vee t = t$. Таким чином, при всіх інтерпретаціях x, y, z дана пропозиція є тавтологією, що і потрібно було довести.</p> 	<p>Вариант 4. Объект x входит в объект y, а y входит в z.</p> <p>Имеем $R_s(x, y) = t$, $R_s(y, z) = t$, $R_s(x, z) = t$, тогда $\neg(t \& t) \vee t = \neg(t \& t) \vee t = f \vee t = t$. Таким образом, при всевозможных интерпретациях x, y, z данное предложение является тавтологией, что и требовалось доказать.</p> 
<p>С. 61.</p>	<p>С. 90–91.</p>

<p>Для аксіоми R.3: $\forall x \forall y (Rs(x, y) \rightarrow \neg Rs(y, x))$.</p> <p>Розглядаємо варіанти структурних відносин двох об'єктів x і y.</p> <p>Варіант 1. Об'єкти x і y структурно незалежні. Тоді $Rs(x, y) = f$, $Rs(y, x) = f$, і $\neg Rs(x, y) \vee Rs(y, x) = \neg f \vee f = t$ $\vee f = t$</p> 	<p>Для аксіомы Rs(3): $\forall x \forall y (Rs(x, y) \rightarrow \neg Rs(y, x))$.</p> <p>Рассматриваем варианты структурных отношений двух объектов x и y.</p> <p>Вариант 1. Объекты x и y независимы друг от друга (имеется в виду структурно). Тогда $Rs(x, y) = f$; $Rs(y, x) = f$ и $\neg Rs(x, y) \vee Rs(y, x) = \neg f \vee f = t$.</p> 
<p>С. 61.</p>	<p>С. 91.</p>
<p>Варіант 2. Один об'єкт включає інший, наприклад x включає y, тоді $Rs(x, y) = f$, $Rs(y, x) = t$ і $\neg Rs(x, y) \vee Rs(y, x) = \neg f \vee t = t$ $\vee f = t$.</p> 	<p>Вариант 2. Один объект включает другой, например x включает y, тогда $Rs(x, y) = f$, $Rs(y, x) = t$ и $\neg Rs(x, y) \vee Rs(y, x) = \neg f \vee t = t$ $\vee f = t$.</p> 
<p>С. 61.</p>	<p>С. 91.</p>
<p>Варіант, коли y включає x, розглядається аналогічно, шляхом перейменування об'єктів.</p> <p>В результаті доведено значимість аксіом R.1 – R.3 в мові α.</p> <p>Як засіб отримання кількісної оцінки структурних відносин об'єктів можна ввести функцію структурної потужності об'єкта, позначивши її функціональним символом Pw. Для цієї функції пропонується наступна система аксіом:</p> <p>Аксіома $Pw.1$: $Pw(x) = 1$, якщо x – примітивний ІО;</p> <p>Аксіома $Pw.2$: $Pw(x) = Pw(x_1) + Pw(x_2)$, якщо $x = \langle NO, \{A\}, (x_1, x_2), \{F\} \rangle$ або, в загальному випадку, $x \in \langle NO, \{A\}, (x_1, x_2), \{F\} \rangle$, де $\{A\}$ і $\{F\}$ можуть бути і порожніми множинами. Знак $+$ розуміється тут як символ додавання натуральних чисел.</p> <p>На підставі аксіом $Pw.1$ і $Pw.2$ покажемо, що функція Pw є адитивною, тобто справедлива теорема 2.1:</p>	<p>Вариант, когда y включает x, рассматривается аналогично, путем переименования объектов.</p> <p>В результате доказана общезначимость аксиом $Rs(1)$-$Rs(3)$ в языке α.</p> <p>В качестве средства получения количественной оценки структурных отношений объектов можно ввести функцию структурной, мощности объекта, обозначив ее функциональным символом Pw. Для этой функции предлагается следующая система аксиом:</p> <ul style="list-style-type: none"> • $Pw(1) Pw(x) = 1$, если x - примитивный ИО; • $Pw(2) Pw(x) = Pw(x_1) + Pw(x_2)$, если $x = \langle NO, \{A\}, (x_1, x_2), BM \rangle$ или, в общем случае, $x \in \langle NO, \{A\}, (x_1, x_2), BM \rangle$, где $\{A\}$ и BM могут быть и пустыми множествами. Знак $+$ понимается здесь как символ суммирования натуральных чисел. Эта операция также может быть введена аксиоматически, что подразумевается известным и детально разработано Д. Гильбертом и П. Бернаисом [259]. <p>На основании аксиом $Pw(1)$ и $Pw(2)$ покажем, что функция Pw является аддитивной, т. е. справедлива теорема 2.1:</p>
<p>С. 62.</p>	<p>С. 91.</p>
$Pw(x) = \sum_{i=1}^n Pw(x_i)$ <p>де $x \in \langle NO, \{A\}, (x_1, x_2, \dots, x_n), \{F\} \rangle$.</p> <p>Для доказу скористаємося методом індукції.</p> <p>Для $i = 1$ маємо: $Pw(x) = Pw(x_1)$, у випадку, якщо x_1 примітивний ІО, то $Pw(x_1) = 1$, і отже, $Pw(x) = 1$. Якщо ж x_1 не примітивний ІО, то повертаємося до початкової посилки.</p> <p>Для $i = 2$ теорема 2.1 справедлива внаслідок аксіом $Pw(2)$. Покажемо, що якщо теорема вірна при $i = n$, то вона вірна і при $i = n + 1$. В цьому випадку $Pw(x) = Pw(\langle NO, \{A\}, (x_1, x_2, \dots, x_n, x_{n+1}), \{F\} \rangle)$.</p>	$Pw(x) = \sum_{i=1}^n Pw(x_i),$ <p>где $x \in \langle NO, \{A\}, (x_1, x_2, \dots, x_n), BM \rangle$.</p> <p>Для доказательства воспользуемся методом индукции.</p> <p>Для $i = 1$ имеем: $Pw(x) = Pw(x_1)$, в случае, если x_1 примитивный ИО, то $Pw(x_1) = 1$, и следовательно, $Pw(x) = 1$. Если же x_1 не примитивный ИО, то возвращаемся к исходнойсылке.</p> <p>Для $i = 2$ теорема 2.1 справедлива вследствие аксиомы $Pw(2)$. Покажем, что если теорема верна при $i = n$, то она верна и при $i = n + 1$. В этом случае $Pw(x) = Pw(\langle NO, \{A\}, (x_1, x_2, \dots, x_n, x_{n+1}), BM \rangle)$.</p>
<p>С. 62.</p>	<p>С. 92.</p>
<p>Оскільки дія функції Pw не зачіпає такі складові об'єкта, як NO, $\{A\}$, $\{F\}$, то можемо ввести новий об'єкт x_n^*, що включає x і x_{n+1}:</p> $x_n^* = \langle NO, \{A\}, (x_n, x_{n+1}), \{F\} \rangle.$ <p>Тоді</p>	<p>Поскольку действие функции Pw не затрагивает такие составляющие объекта, как NO, $\{A\}$, BM, то можем ввести новый объект x_n^*, включающий x и x_{n+1}:</p> $x_n^* = \langle NO, \{A\}, (x_n, x_{n+1}), BM \rangle.$ <p>Тогда</p>

<p>$Pw(x) = Pw(\langle NO, \{A\}, (x_1, x_2, \dots, x_n^*), \{F\} \rangle) =$ $= \sum_{i=1}^n Pw(x_i) = Pw(x_1) + Pw(x_2) + \dots + Pw(x_n^*) =$ $= Pw(x_1) + Pw(x_2) + \dots + Pw(\langle NO, \{A\}, (x_n, x_{n+1}), \{F\} \rangle) =$ $= Pw(x_1) + Pw(x_2) + \dots + Pw(x_n) + Pw(x_{n+1})$ по аксіомі Pw.2. Отже, теорема 2.1 вірна і для довільного натурального числа n. Далі покажемо, що справедлива теорема 2.2: Функція Pw(x) повертає кількість примітивних об'єктів, що містяться в інформаційному об'єкті X.</p>	<p>$Pw(x) = Pw(\langle NO, \{A\}, (x_1, x_2, \dots, x_n^*), BM \rangle) = \sum_{i=1}^n Pw(x_i) = Pw(x_1) +$ $+ Pw(x_2) + \dots + Pw(x_n^*) = Pw(x_1) + Pw(x_2) + \dots +$ $+ Pw(\langle NO, \{A\}, (x_n, x_{n+1}), BM \rangle) =$ $= Pw(x_1) + Pw(x_2) + \dots + Pw(x_n) + Pw(x_{n+1})$. по аксіомі 2. Следовательно, теорема 2.1 верна и для произвольного натурального числа n. Далее покажем, что справедлива теорема 2.2: Функция Pw(x) возвращает количество примитивных объектов, содержащихся в информационном объекте X.</p>
<p>С. 63.</p>	<p>С. 92.</p>
<p>Скористаємося індукцією по глибині індекса вкладення об'єктів I. Нехай I = 1, тоді $\sum_{i=1}^n Pw(x_i) = n$, якщо всі x_i що входять в X є примітивними ІО. Покажемо, що якщо теорема справедлива при глибині індекса вкладення k, то вона справедлива і при k+1. При глибині вкладення, яка дорівнює k маємо:</p> $\sum_{i=1}^n Pw_k(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k}) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_k$ <p>коли всі $x_{l_1 l_2 \dots l_k}$ є примітивними ІО і $I_k = Pw(x_{l_1 l_2 \dots l_k})$. При глибині вкладення k+1 маємо:</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} \sum_{l_{k+1}=1}^{n_{k+1}} Pw(x_{l_1 l_2 \dots l_k l_{k+1}})$	<p>Воспользуемся индукцией по глубине индекса вложения объектов I. Пусть I = 1, тогда $Pw(x) = \sum_{i=1}^n Pw(x_i) = n$, если все x_i, входящие в X, являются примитивными ИО. Покажем, что если теорема справедлива при глубине индекса вложения k, то она справедлива и при k + 1. При глубине вложения равной k имеем:</p> $Pw_k(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k}) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_k$ <p>когда все $x_{l_1 l_2 \dots l_k}$ являются примитивными ИО и $I_k = Pw(x_{l_1 l_2 \dots l_k})$. При глубине вложения k + 1 имеем:</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} \sum_{l_{k+1}=1}^{n_{k+1}} Pw(x_{l_1 l_2 \dots l_k l_{k+1}})$
<p>С. 63.</p>	<p>С. 92.</p>
<p>Відповідно до теореми 2.1 можемо цю суму розкрити як</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} (Pw(x_{l_1 l_2 \dots l_k l_{k+1}}) + Pw(x_{l_1 l_2 \dots l_k l_{k+1}}))$	<p>В соответствии с теоремой 2.1 можем эту сумму раскрыть как</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} ((Pw(x_{l_1 l_2 \dots l_k l_{k+1}}) + Pw(x_{l_1 l_2 \dots l_k l_{k+1}})) + \dots + Pw(x_{l_1 l_2 \dots l_k l_{k+1}}))$
<p>С. 63.</p>	<p>С. 92–93.</p>
<p>Розкриваючи дужки, отримаємо:</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k l_{k+1}}) +$ $+ \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k l_{k+1}}) + \dots +$ $+ \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k l_{k+1}})$	<p>Раскрывая скобки, получим:</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k l_{k+1}}) +$ $+ \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k l_{k+1}}) + \dots +$ $+ \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} Pw(x_{l_1 l_2 \dots l_k l_{k+1}})$
<p>С. 64.</p>	<p>С. 93.</p>
<p>Для кожного з цих доданків, виходячи із заданої передумови теореми, отримуємо:</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1} + \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1} + \dots + \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1} =$ $\sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1}$ <p>що і потрібно було довести.</p>	<p>Для каждого из этих слагаемых, исходя из заданной предпосылки теоремы, получаем:</p> $Pw_{k+1}(x) = \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1} + \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1} + \dots + \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1} =$ $= \sum_{l_1=1}^{n_1} \sum_{l_2=1}^{n_2} \dots \sum_{l_k=1}^{n_k} I_{k+1}$ <p>что и требовалось доказать.</p>
<p>С. 64.</p>	<p>С. 93.</p>
<p>Для забезпечення замкнутості числення об'єктів на множині M необхідно поставити умову, щоб результат дії функціонального символу Pw також був об'єктом. Тому замість безпосереднього використання натурального числа вважатимемо, що Pw(x) = <NO, A> – однопараметричний об'єкт, де</p>	<p>Для обеспечения замкнутости исчисления объектов на множестве M необходимо потребовать, чтобы результат действия функционального символа Pw также был объектом. Поэтому вместо непосредственного использования натурального числа будем считать, что Pw(x) = < NO, A > - одно-</p>

	<p>$A = \langle NA, \{N\}, a \rangle$, де $\{N\}$ – множина натуральних чисел; $a \in \{N\}$. У свою чергу, оскільки мова α допускає побудову рекурсивних термів, формально припустимим є терм $Pw(Pw(x))$. Тому $Pw(Pw(x)) = Pw(\langle NO, A \rangle)$ покладемо рівним x_0, де $x_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$, тобто порожній об'єкт. Порожній об'єкт x_0 необхідно замкнути сам на себе і вважати, що $Pw(x_0) = x_0$, тоді рекурсія виду $Pw(Pw\dots Pw(x)\dots)$ завжди дасть x_0. З міркувань суворості доведеться доповнити систему аксіом для відношення R_s:</p> <p>Аксіома R.1.1: $\forall x(\neg R_s(x_0, x)) \vee \forall x(\neg R_s(x_0, x))$; Аксіома R.1.2: $\forall x(\neg R_s(x, x_0)) \vee \forall x(\neg R_s(x, x_0))$. Обґрунтування того, що аксіоми R.1.1 і R.1.2 є тавтологіями, абсолютно аналогічно обґрунтуванню аксіоми R.1.</p>	<p>параметрический объект, где $A = \langle NA, \{\text{множество_натуральных_чисел}\}, a \rangle$, где $a \in \{\text{множество_натуральных_чисел}\}$. В свою очередь, поскольку язык α допускает построение рекурсивных термов, формально допустим терм $Pw(Pw(x))$. Поэтому $Pw(Pw(x)) = Pw(\langle NO, A \rangle)$ положим равным x_0, где $x_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$, т. е. пустой объект. Пустой объект x_0 необходимо замкнуть сам на себя и считать, что $Pw(x_0) = x_0$, тогда рекурсия вида $Pw(Pw\dots Pw(x)\dots)$ всегда даст x_0. Из соображений строгости придется дополнить систему аксиом для отношения R_s:</p> <ul style="list-style-type: none"> • $R_s(00) \vee \forall x(\neg R_s(x_0, x))$; • $R_s(01) \vee \forall x(\neg R_s(x, x_0))$. <p>Обоснование того, что аксиомы $R_s(00)$ и $R_s(01)$ являются тавтологиями, совершенно аналогично обоснованию аксиомы $R_s(1)$.</p>
С. 65.		С. 93–94.
	<p>Використання числення предикатів з рівністю примушує нас ввести поняття рівності об'єктів. Насправді, трактувати його необхідно як еквівалентність об'єктів. Вважатимемо, що два об'єкти рівні (еквівалентні один одному), якщо</p> $x = \langle NA_x, \{A\}_x, \{O\}_x, \{F\}_x \rangle;$ $y = \langle NA_y, \{A\}_y, \{O\}_y, \{F\}_y \rangle;$ $\{A\}_x = \{A\}_y, \{O\}_x = \{O\}_y, \{F\}_x = \{F\}_y.$ <p>Імена об'єктів при цьому можуть бути різними.</p>	<p>Использование ИП с равенством заставляет нас ввести понятие равенства объектов. На самом деле, трактовать его необходимо как эквивалентность объектов. Будем считать, что два объекта равны (эквивалентны друг другу), если</p> $x = \langle NA_x, \{A\}_x, \{O\}_x, BM_x \rangle;$ $y = \langle NA_y, \{A\}_y, \{O\}_y, BM_y \rangle;$ $\{A\}_x = \{A\}_y, \{O\}_x = \{O\}_y, BM_x = BM_y.$ <p>Имена объектов при этом могут быть различными.</p>
С. 65.		С. 94.
	<p>Очевидно, що далі потрібно визначити відношення рівності для множин атрибутів, вкладених об'єктів і моделей функціонування. Дві множини атрибутів $\{A\}_x$ і $\{A\}_y$ вважатимемо рівними (еквівалентними), якщо для всіх $A_i \in \{A\}_x$ і $A_j \in \{A\}_y$ спостерігається попарна рівність при однаковому значенні індекса i, тобто $A_{ix} = A_{iy}$, де $i = 1, 2 \dots n$, при цьому n – потужність множин $\{A\}_x$ і $\{A\}_y$.</p> <p>Два атрибути A_ϕ і A_ψ назвемо рівними (еквівалентними), якщо $NA_\phi = NA_\psi$, $SA_\phi = SA_\psi$, $VA_\phi = VA_\psi$.</p> <p>Рівність списків вкладених об'єктів $\{O\}_x$ і $\{O\}_y$ визначається аналогічно рівності множин атрибутів, тобто $\{O\}_x = \{O\}_y$, якщо потужності множин $\{O\}_x$ і $\{O\}_y$ рівні і кожен $O_i x = O_i y$. Очевидно, що це визначення рекурсивне, оскільки об'єкти $O_i x$ і $O_i y$ можуть мати свої вкладені об'єкти.</p>	<p>Очевидно, что далее требуется определить отношение равенства для множеств атрибутов, вложенных объектов и моделей поведения. Два множества атрибутов $\{A\}_x$ и $\{A\}_y$ будем считать равными (эквивалентными), если для всех $A_i \in \{A\}_x$ и $A_j \in \{A\}_y$ наблюдается попарное равенство при одинаковом значении индекса i, т. е. $A_{ix} = A_{iy}$, где $i = 1, 2 \dots, n$, при этом n — мощность множеств $\{A\}_x$ и $\{A\}_y$.</p> <p>Два атрибута A_ϕ и A_ψ назовем равными (эквивалентными), если $NA_\phi = NA_\psi$, $SA_\phi = SA_\psi$, $VA_\phi = VA_\psi$.</p> <p>Равенство списков вложенных объектов $\{O\}_x$ и $\{O\}_y$ определяется аналогично равенству множеств атрибутов, т. е. $\{O\}_x = \{O\}_y$, если мощности множеств $\{O\}_x$ и $\{O\}_y$ равны и каждый $O_{ix} = O_{iy}$. Очевидно, что это определение рекурсивно, так как объекты O_{ix} и O_{iy} могут иметь свои вложенные объекты.</p>
С. 65–66.		С. 94.
	<p>Необхідно також ввести поняття потужності об'єкта Pp, тобто функції, що показує кількість вкладених об'єктів.</p> <p>Введемо наступні аксіоми:</p> <p>Аксіома $Pp.1$: $Pp(x) = \emptyset$, якщо x – примітивний ІО;</p> <p>Аксіома $Pp.2$: $Pp(x) = k$, якщо $x = \langle NO, \{A\}, (x_1, x_2, \dots, x_k), \{F\} \rangle$, де $\{A\}$ і $\{F\}$ можуть бути і порожніми множинами.</p> <p>Можна ввести $Pp(x)$ і конструктивно, тобто використовувати продукційні правила для обчислення k, оскільки це число в явному вигляді може бути і невідоме. За аналогією з функцією Pw вважатимемо, що $Pp(x) = \langle NO, A \rangle$ – однопараметричний об'єкт, де $A = \langle NA, \{N\}, a \rangle$, де $a \in \{N\}$. $Pp(x_0)$ покладемо рівним x_0.</p>	<p>Придется ввести понятие первичное мощности объекта Pp, т. е. функции, показывающей количество вложенных объектов.</p> <p>Введем следующие аксиомы:</p> <ul style="list-style-type: none"> • $Pp(1) Pp(x) = \emptyset$, если x - примитивный ИО; • $Pp(2) Pp(x) = k$, если $x = \langle NO, \{A\}, (x_1, x_2, \dots, x_k), BM \rangle$, где $\{A\}$ и BM могут быть и пустыми множествами. <p>Можно ввести $Pp(x)$ и конструктивно, т. с. использовать продукционные правила для вычисления k, поскольку это число в явном виде может быть и неизвестно. По аналогии с функцией Pw будем считать, что $Pp(x) = \langle NO, A \rangle$ - однопараметрический объект, где $A = \langle NA, \{\text{множество_натуральных_чисел}\}, a \rangle$, где $a \in \{\text{множество_натуральных_чисел}\}$.</p>

		ство_натуральных_чисел}. Рр(хд) положим равным х0.					
С. 66.		С. 94–95.					
<p>Розширення алгебраїчної системи Ф можна здійснити шляхом введення нових відношень над ІО (відповідно нових предикативних символів в мову α), що дозволяє заздалегідь не обмежувати спільність системи Ф. При цьому пропонується наступна схема введення нових відношень:</p> <table border="1"> <thead> <tr> <th>Бинарне відношення</th> <th>Символ відношення</th> <th>Аксиоми відношення</th> </tr> </thead> <tbody> <tr> <td>< ім'я відношення ></td> <td>< знак, який позначає відношення у формальній системі ></td> <td>< список аксіом ></td> </tr> </tbody> </table> <p>Слюняєв із тексту зробив таблицю. Плагіат.</p>	Бинарне відношення	Символ відношення	Аксиоми відношення	< ім'я відношення >	< знак, який позначає відношення у формальній системі >	< список аксіом >	<p>Расширение алгебраической системы Ф можно осуществить путем введения новых отношений над ІО (соответственно новых предикатных символов в язык α), что позволяет не ограничивать заранее общность системы Ф. При этом предлагается следующая схема введения новых отношений:</p> <p>Отношение_бинарное:: = < имя_отношения >; Символ_отношения:: = < знак_обозначающий_отношение_в_формальной_системе >; Аксиомы_отношения:: " < список_аксиом >.</p>
Бинарне відношення	Символ відношення	Аксиоми відношення					
< ім'я відношення >	< знак, який позначає відношення у формальній системі >	< список аксіом >					
С. 66.		С. 95.					
<p>Оскільки найбільш поширені відношення мають типові властивості, то можна запропонувати сукупність аксіом, визначених апріорно і відомих формальній системі, щоб згодом задавати їх просто посиланнями. Запишемо їх як</p> <p>Аксиома А.1: $\forall x \forall y (xRy \rightarrow yRx)$; Аксиома А.2: $\forall x \forall y (xRy \rightarrow \neg(yRx))$; Аксиома А.3: $\forall x (xRx)$; Аксиома А.4: $\forall x \neg(xRx)$; Аксиома А.5: $\forall x \forall y \forall z ((xRy) \& (yRz) \rightarrow xRz)$.</p> <p>Тут аксіома А.1 виражає властивість симетричності, аксіома А.2 – асиметричності, аксіома А.3 – рефлексивності, аксіома А.4 – антирефлексивності і аксіома А.5 – транзитивності.</p> <p>Задаючи таким чином нові відношення (предикативні символи), ми можемо будувати різні, довільні до деякої міри аксіоматизовані системи Ф в мові числення предикатів.</p>	<p>Поскольку наиболее распространенные отношения имеют типичные свойства, то можно предложить совокупность аксиом, predeterminedных по умолчанию, априорно известных формальной системе, чтобы впоследствии задавать их просто ссылками. Запишем их как</p> <p>АО\emptyset1 $\forall x \forall y (xRy \rightarrow yRx)$; АО$\emptyset$2 $\forall x \forall y (xRy \rightarrow \neg(yRx))$; АО$\emptyset$3 $\forall x (xRx)$; АО\emptyset4 $\forall x \neg(xRx)$; АО\emptyset5 $\forall x \forall y \forall z ((xRy) \& (yRz) \rightarrow xRz)$.</p> <p>Здесь АО$\emptyset$1 выражает свойство симметричности, АО\emptyset2 - асимметричности, АО\emptyset3 - рефлексивности, АО\emptyset4 - антирефлексивности и АО\emptyset5 - транзитивности.</p> <p>Задавая таким образом новые отношения (предикатные символы), мы можем строить различные, произвольные до некоторой степени аксиоматизируемые системы Ф в языке исчисления предикатов.</p>						
С. 67.		С. 95.					
<p>Відношення рівності (еквівалентності) об'єктів, очевидно, вимагає детальнішого розгляду внаслідок складної структури ІО. Введемо наступні визначення. Два інформаційні об'єкти x і y назовемо одновидовими, якщо $N_{Ox} \neq N_{Oy}$, $\{A\}x = \{A\}y$, $\{O\}x = \{O\}y$, $\{F\}x = \{F\}y$, тобто вони еквівалентні з точністю до власних імен об'єктів. Відношення одновидової еквівалентності є симетричним, антирефлексивним і транзитивним (задовольняє аксіомам А.1, А.4, А.5).</p> <p>Введемо еквівалентність з точністю до імен атрибутів.</p> <p>Два ІО x і y називаються об'єктами одного класу, якщо $N_{Ox} \neq N_{Oy}$, $\{O\}x = \{O\}y$, $\{F\}x = \{F\}y$ і для всіх i $N_{Axi} = N_{Ayi}$, $S_{Axi} = S_{Ayi}$, де $i=1,2,\dots,k$, де k – число атрибутів в множині $\{A\}x$ і $\{A\}y$ (потужність $\{A\}x = \{A\}y$). Відношення класової еквівалентності також є симетричним, антирефлексивним і транзитивним.</p>	<p>Отношение равенства (эквивалентности) объектов, очевидно, требует более детального рассмотрения вследствие сложной структуры ІО. Введем следующие определения. Два информационных объекта x и y назовем информационными объектами одного вида (одновидовыми объектами), если $N_{Ox} \neq N_{Oy}$, $\{A\}x = \{A\}y$, $\{O\}x = \{O\}y$, $\{F\}x = \{F\}y$, т. е. они эквивалентны с точностью до собственных имен объектов. Отношение одновидової эквивалентности является симметричным, антирефлексивным и транзитивным (удовлетворяет аксиомам АО\emptyset1, АО\emptyset4, АО\emptyset5).</p> <p>Введем эквивалентность с точностью до имен атрибутов.</p> <p>Два ІО x и y называются объектами одного класса (классовая эквивалентность), если $N_{Ox} \neq N_{Oy}$, $\{O\}x = \{O\}y$, $\{F\}x = \{F\}y$ и для всех i $N_{Axi} = N_{Ayi}$, $S_{Axi} = S_{Ayi}$, где $i = 1, 2, \dots, k$, где k - число атрибутов в множествах $\{A\}x$ и $\{A\}y$ (мощность $\{A\}x = \{A\}y$). Отношение классовой эквивалентности также является симметричным, антирефлексивным и транзитивным.</p>						
С. 67.		С. 96.					
<p>Якщо два об'єкти знаходяться у відношенні одновидової еквівалентності, то вони обов'язково знаходяться і у відношенні класової еквівалентності, тобто справедлива наступна теорема.</p>	<p>Если два объекта находятся в отношении одновидової эквивалентности (ОЭ), то они обязательно находятся и в отношении классовой эквивалентности (КЭ), т. е. справедлива следующая теорема.</p>						

	<p>Теорема 2.3. $\forall x \forall y \forall z ((xRy \vee yRz) \rightarrow (xRz))$.</p> <p>Дана теорема безпосередньо виходить з визначень одновидової та класової еквівалентності, оскільки одновидова еквівалентність є окремим випадком класової еквівалентності при $\forall x i = \forall Ay i$.</p> <p>Зворотнє твердження, природно, є невірним з тих же причин. Тут через R_0 позначаємо одновидову еквівалентність, а через R_k – класову еквівалентність.</p>	<p>Теорема 2.3. $\forall x \forall y ((xR_0y) \rightarrow (xR_ky))$.</p> <p>Данная теорема непосредственно следует из определений ОЭ и КЭ, так как ОЭ является частным случаем КЭ при $\forall Ax i = \forall Ay i$.</p> <p>Обратное утверждение, естественно, неверно по тем же причинам. Здесь через R_0 обозначаем одновидовую эквивалентность, а через R_k — классовую эквивалентность.</p>
С. 67–68.		С. 96.
	<p>Два IO x і y називаються структурно еквівалентними, якщо $NOx \neq NOy$, $\{O\}x = \{O\}y$, $\{F\}x = \{F\}y$, $\{A\}x \equiv \{A\}y$, де знак \equiv позначає структурну еквівалентність. СЕ для $\{A\}$ означає, що потужності множин $\{A\}x$ і $\{A\}y$ однакові і $SAx_i = SAy_i$ при $i = 1, 2, \dots, k$, де k – потужність $\{A\}x$ і $\{A\}y$ відповідно. Таким чином, співпадають тільки визначення атрибутів, а імена і значення їх можуть бути різні. Для множин $\{O\}x$ і $\{O\}y$ структурна еквівалентність означає, що первинні потужності $Pp(x)$ і $Pp(y)$ рівні і $Ox_i = Oy_i$, де $i = 1, 2, \dots, m$, де $m = Pp(x) = Pp(y)$.</p> <p>Відношення структурної еквівалентності є симетричним, антирефлексивним і транзитивним.</p>	<p>Два IO x и y называются структурно эквивалентными (СЭ), если $N ox \neq N Oy$, $\{O\}x \equiv \{O\}y$, $BMx = BMy$, $\{A\}x \equiv \{A\}y$, где знак \equiv обозначает структурную эквивалентность. СЭ для $\{A\}$ означает, что мощности множеств $\{A\}x$ и $\{A\}y$ одинаковы и $SAx_i = SAy_i$ при $i = 1, 2, \dots, k$, где k - мощность $\{A\}x$ (соответственно? и $\{A\}y$). Таким образом, совпадают только определения атрибутов, а имена и значения их могут быть различны. Для множеств $\{O\}x$ и $\{O\}y$ структурная эквивалентность обозначает, что первичная мощность $Pp(x)$ и $Pp(y)$ равны и $Ox_i = Oy_i$, где $i = 1, 2, \dots, m$, где $m = Pp(x) = Pp(y)$.</p> <p>Дополнительное разъяснение структурной эквивалентности для примитивных, структурных и однопараметрических объектов не требуется, так как они подпадают под общее определение данное выше. Исходя из такого определения, классовая эквивалентность не является частным случаем структурной эквивалентности, поскольку в случае СЭ вложенные объекты находятся в отношении СЭ, а не эквивалентности, как в случае КЭ. Отношение СЭ является симметричным, антирефлексивным и транзитивным.</p>
С. 68.		С. 96.
	<p>Функція визначення атрибута.</p> <p>Оскільки при аналізі процесів функціонування буде потрібно порівняння різних характеристик IO, введемо функцію визначення атрибута $I(p, q)$, де p – ім'я IO, q – ім'я атрибута.</p> $I(p, q) = \begin{cases} x_0, & \text{якщо } \forall N_A (N_A \in p) \wedge (N_A \neq q) \\ 0 = \langle p, N_A, A_q \rangle & \text{– у протилежному випадку} \end{cases}$ <p>Слюняєв не зміг перекласти українською слово «извлечения» – написав «визначення», а російське слово «извлекаемого» взагалі вирішив не перекладати. Смішний плагіат.</p>	<p>Функция извлечения атрибута.</p> <p>Поскольку при анализе поведения потребуется сравнение различных характеристик IO, введем функцию извлечения атрибута $I(p, q)$, где p - имя IO, q - имя извлекаемого атрибута.</p> $I(p, q) = \begin{cases} x_0, & \text{если } \forall N_A ((N_A \in p) \& (N_A \neq q)), \\ 0 = \langle p, N_A, A_q \rangle & \text{– в противном случае.} \end{cases}$
С. 68.		С. 97.
	<p>Це означає, що у разі відсутності в множині $\{A\}$ атрибута з ім'ям q, функція приймає значення порожнього інформаційного об'єкта x_0. Виходить, що функція $I(p, q)$ приймає значення на множині $x_0 \cup \{Oop\}$, де $\{Oop\}^*$ – множина усіх можливих однопараметричних IO. Необхідно зазначити апріорно, що взяття функції I від однопараметричного об'єкта дає той же самий однопараметричний об'єкт: $I(p, q) = p$, якщо p – однопараметричний IO.</p> <p>Це дозволяє надалі ввести необхідні математичні операції над однопараметричними IO (ОПІО). Почнемо цей розгляд з операції арифметичного підсумовування.</p> <p>Нехай $\Sigma(X, Y) = Z$, де $X = \langle p, A \rangle$, $Y = \langle q, B \rangle$ є ОПІО з іменами p і q, а також атрибутами A і B, відповідно, при цьому $A = \langle NA, SA, VA \rangle$, $B = \langle NB, SB, VB \rangle$.</p>	<p>Это означает, что в случае отсутствия в множестве $\{A\}$ атрибута с именем q, функция принимает значение пустого информационного объекта x_0. Получается, что функция $I(p, q)$ принимает значения на множестве $x_0 \cup \{Oop\}$, где $\{Oop\}^*$ - множество всевозможных однопараметрических IO. Необходимо постулировать, что взятие функции I от однопараметрического объекта дает тот же самый однопараметрический объект: $I(p, q) = p$, если p — однопараметрический IO.</p> <p>Это позволяет в дальнейшем ввести необходимые математические операции над однопараметрическими IO (ОПИО). Начнем это рассмотрение с операции арифметического суммирования.</p> <p>Пусть $\Sigma(X, Y) = Z$, где $X = \langle p, A \rangle$, $Y = \langle q, B \rangle$ есть ОПИО с именами p и q и атрибутами A и B, соответственно, при этом $A = \langle NA, SA, VA \rangle$, $B =$</p>

	<p>Введемо наступні аксіоми:</p> <p>Аксіома В.1: $\forall x((x = x_0) \rightarrow (\sum(x, y) = x_0))$;</p> <p>Аксіома В.2: $\forall x((x = x_0) \rightarrow (\sum(y, x) = x_0))$.</p>	<p><NB, SB, VB></p> <p>Введем следующие аксиомы:</p> <ul style="list-style-type: none"> • $\Lambda\Sigma01: \forall x((x = x_0) \rightarrow (\sum(x, y) = x_0))$; • $\Lambda\Sigma02: \forall x((x = x_0) \rightarrow (\sum(y, x) = x_0))$.
С. 69.		С. 97.
	<p>В результаті логічного виводу отримаємо, що $\Sigma(x_0, x_0) = x_0$. Об'єкт x_0 служить ознакою виникнення помилки.</p> <p>Аксіома В.3: $\forall x\forall y(\neg(S_A = S_B) \rightarrow (\sum(X, Y) = x_0))$;</p> <p>Аксіома В.4: $\forall x\forall y\forall z((S_A = S_B) \& (V_C \in S_A) \rightarrow (\sum(X, Y) = Z))$;</p> <p>де $Z = \langle p-q, \Sigma A, B \rangle$;</p> <p>де $\Sigma A, B = C = \langle N_C, S_C, V_C \rangle$, $N_C = N_A - N_B$, $S_C = S_A = S_B$, $V_C = \Sigma V_A, V_B$;</p> <p>Аксіома В.5: $\forall x\forall y\forall z((S_A = S_B) \& \neg(V_C \in S_A) \rightarrow (\sum(X, Y) = x_0))$.</p>	<p>В результате логического вывода получим, что $\Sigma(x_0, x_0) = x_0$. Объект x_0 служит признаком возникновения ошибки;</p> <ul style="list-style-type: none"> • $\Lambda\Sigma03: \forall x\forall y(\neg(S_A = S_B) \rightarrow (\sum(X, Y) = x_0))$; • $\Lambda\Sigma04: \forall x\forall y\forall z((S_A = S_B) \& (V_C \in S_A) \rightarrow (\sum(X, Y) = Z))$, где $Z = \langle p-q, \Sigma A, B \rangle$, где $\Sigma A, B = C = \langle N_C, S_C, V_C \rangle$, $N_C = N_A - N_B$, $S_C = S_A = S_B$, $V_C = \Sigma V_A, V_B$; • $\Lambda\Sigma05: \forall x\forall y\forall z((S_A = S_B) \& \neg(V_C \in S_A) \rightarrow (\sum(X, Y) = x_0))$.
С. 69.		С. 97.
	<p>Це випадок, коли V_C лежить за межами множини S_A.</p> <p>$V_C = \Sigma V_A, V_B$ розуміється як арифметична сума, яка може бути побудована конструктивно для відповідних видів чисел.</p> <p>Операцію можна зробити і n-арною, використовуючи рекурсивне визначення:</p> $\sum_{i=1}^n x_i = \sum \left(\sum_{i=1}^{n-1} x_i \right), x_n.$	<p>Это случай, когда V_C лежит за пределами множества S_A.</p> <p>$V_C = \Sigma V_A, V_B$ понимается как арифметическая сумма, которая может быть построена конструктивно для соответствующих видов чисел.</p> <p>Операцию Σ можно сделать и n-арной, используя рекурсивное определение:</p> $\sum_{i=1}^n (x_i) = \sum \left(\sum_{i=1}^{n-1} x_i \right), x_n.$
С. 69.		С. 97–98.
	<p>Легко відмітити, що для введеного таким чином додавання ОПІО не виконується комутативний закон: $\Sigma(X, Y) \neq \Sigma(Y, X)$, оскільки результатом лівої суми є $Z = \langle p-q, \Sigma A, B \rangle$, а результатом правої суми $Z' = \langle q-p, \Sigma B, A \rangle$.</p> <p>Та зате для додавання ОПІО виконується асоціативний закон</p> $\Sigma(X, Y, Z) = \Sigma(X, \Sigma(Y, Z)) = \Sigma(\Sigma(X, Y), Z).$	<p>Легко заметить, что для введенного таким образом сложения ОПИО не выполняется коммутативный закон: $\Sigma(X, Y) \neq \Sigma(Y, X)$, так как результатом левой суммы является $Z = \langle p-q, \Sigma A, B \rangle$, а результатом правой суммы - $Z' = \langle q-p, \Sigma B, A \rangle$.</p> <p>Но зато для сложения ОПИО выполняется ассоциативный закон</p> $\Sigma(X, Y, Z) = \Sigma(X, \Sigma(Y, Z)) = \Sigma(\Sigma(X, Y), Z).$
С. 69.		С. 98.
	<p>З рекурсивного визначення суми отримуємо: $\Sigma(X, Y, Z) = \Sigma(X, \Sigma(Y, Z))$.</p> <p>Покажем, что это в точности равно $\Sigma(\Sigma(X, Y), Z)$</p> <p>Нехай $X = \langle p, A \rangle$, $Y = \langle q, B \rangle$, $Z = \langle l, C \rangle$, тоді $\Sigma(X, \Sigma(Y, Z))$ набуває вигляду $\Sigma(X, \Sigma(Y, Z)) = \Sigma(X, \langle q-l, \Sigma B, C \rangle) = \Sigma(X, \langle q-l, \langle N_B - N_C, S_B, \Sigma V_B, V_C \rangle \rangle)$</p> <p>У свою чергу</p> $\Sigma(\Sigma(X, Y), Z) = \Sigma \langle p-q, \Sigma A, B \rangle, Z = \Sigma \langle p-q, N_A - N_B, S_A, \Sigma V_A, V_B \rangle, Z = \langle p-q-l, N_A - N_B - N_C, S_A, \Sigma(\Sigma V_A, V_B), V_C \rangle$	<p>Из рекурсивного определения суммы получаем: $\Sigma(X, Y, Z) = \Sigma(X, \Sigma(Y, Z))$.</p> <p>Покажем, что это в точности равно $\Sigma(\Sigma(X, Y), Z)$</p> <p>Пусть $X = \langle p, A \rangle$, $Y = \langle q, B \rangle$, $Z = \langle l, C \rangle$, тогда $\Sigma(X, \Sigma(Y, Z))$ принимает вид $\Sigma(X, \Sigma(Y, Z)) = \Sigma(X, \langle q-l, \Sigma B, C \rangle) = \Sigma(X, \langle q-l, \langle N_B - N_C, S_B, \Sigma V_B, V_C \rangle \rangle) = \langle p-q-l, \langle N_A - N_B - N_C, S_A, \Sigma V_A, \Sigma V_B, V_C \rangle \rangle$.</p> <p>В свою очередь,</p> $\Sigma(\Sigma(X, Y), Z) = \Sigma \langle p-q, \Sigma A, B \rangle, Z = \Sigma \langle p-q, \langle N_A - N_B, S_A, \Sigma V_A, V_B \rangle \rangle, Z = \langle p-q-l, \langle N_A - N_B - N_C, S_A, \Sigma(\Sigma V_A, V_B), V_C \rangle \rangle$
С. 70.		С. 98.
	<p>Оскільки V_A, V_B, V_C – це числа, то для них асоціативний закон виконується і $\Sigma V_A, \Sigma V_B, V_C = \Sigma(\Sigma V_A, V_B), V_C$. Таким чином, доведена справедливість асоціативного закону для ОПІО.</p> <p>Покажемо справедливості закону нульової множини для складання ОПІО.</p> <p>Теорема 2.4. Для всіх ОПІО виконується суворі рівність (еквівалентність):</p> $\sum \left(\left(\sum_{i=1}^{n-1} x_i \right), x_0 \right) = x_0$ <p>Слюняєв пише «Таким чином, доведена», нібито це він щось доводив, а насправді переписує російську дисертацію. Плагіат.</p>	<p>Поскольку V_A, V_B, V_C - это числа, то для них ассоциативный закон выполняется и $\Sigma V_A, \Sigma V_B, V_C = \Sigma(\Sigma V_A, V_B), V_C$. Таким образом, доказана справедливость ассоциативного закона для ОПИО.</p> <p>Покажем справедливость закона нулевого множества для сложения ОПИО.</p> <p>Теорема 2.4. Для всех ОПИО выполняется строгое равенство (эквивалентность):</p> $\sum \left(\left(\sum_{i=1}^{n-1} x_i \right), x_0 \right) = x_0.$
С. 70.		С. 98.

<p>Для доказу скористаємося індукцією по довжині суми. При $n = 2$ маємо $\Sigma(x), x_0 = x_0$ згідно Аксиоми В.2. Припустимо, що наше твердження справедливе при довільному n, і покажемо, що з цього виходить його істинність при $n + 1$:</p> $\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_0\right) = \Sigma\left(\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_{n+1}\right), x_0\right)$ <p>Розглянемо два можливі випадки:</p> <p>а) сума $\sum_{i=1}^{n-1} x_i$ дає деякий ОПЮ, який позначимо як M. Тоді отримуємо:</p> $\Sigma(\Sigma(M, x_{n+1}), x_0) = \Sigma(M, \Sigma(x_{n+1}, x_0)),$ $\Sigma\left(\Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_{n+1}\right), x_0\right) = \Sigma(\Sigma(M, x_{n+1}), x_0)$	<p>Для доказательства воспользуемся индукцией по длине суммы. При $n = 2$ имеем $\Sigma(x), x_0 = x_0$, согласно АΣ02. Положим, что наше утверждение справедливо при произвольном n, и покажем, что из этого следует его истинность при $n + 1$:</p> $\Sigma\left(\left(\sum_{i=1}^{n+1} x_i\right), x_0\right) = \Sigma\left(\Sigma\left(\left(\sum_{i=1}^n x_i\right), x_{n+1}\right), x_0\right).$ <p>Рассмотрим два возможных случая:</p> <p>а) сумма $\sum_{i=1}^n x_i$ дает некоторый ОПЮ, который обозначим как M. Тогда получаем:</p> $\Sigma(\Sigma(M, x_{n+1}), x_0) = \Sigma(M, \Sigma(x_{n+1}, x_0)),$ $\Sigma\left(\Sigma\left(\left(\sum_{i=1}^n x_i\right), x_{n+1}\right), x_0\right) = \Sigma(\Sigma(M, x_{n+1}), x_0)$
<p>С. 70.</p>	<p>С. 99.</p>
<p>згідно з асоціативним законом. За аксіомою В.2 отримуємо $\Sigma(x_{n+1}, x_0)$ і тоді</p> $\Sigma(M, x_0) = \Sigma\left(\sum_{i=1}^n x_i\right), x_0 = x_0$ <p>за послідовної індукції:</p> <p>б) сума $\sum_{i=1}^{n-1} x_i$ дає в результаті x_0. Тоді</p>	<p>согласно ассоциативному закону. По АΣ02 получаем $\Sigma(x_{n+1}, x_0) = x_0$, и тогда $\Sigma(M, x_0) = \Sigma\left(\sum_{i=1}^n x_i\right), x_0 = x_0$ по послідовній індукції:</p> <p>б) сума $\sum_{i=1}^{n-1} x_i$ дає в результаті x_0. Тоді</p>
<p>С. 71.</p>	<p>С. 99.</p>
$\Sigma\left(\left(\sum_{i=1}^{n+1} x_i\right), x_0\right) = \Sigma\left(\Sigma\left(\left(\sum_{i=1}^n x_i\right), x_{n+1}\right), x_0\right) = \Sigma(\Sigma(x_0, x_{n+1}), x_0)$ <p>За аксіомою В.1 отримуємо:</p> $\Sigma(\Sigma(x_0, x_{n+1}), x_0) = \Sigma(x_0, x_0) = x_0$	$\Sigma\left(\left(\sum_{i=1}^{n+1} x_i\right), x_0\right) = \Sigma\left(\Sigma\left(\left(\sum_{i=1}^n x_i\right), x_{n+1}\right), x_0\right) = \Sigma(\Sigma(x_0, x_{n+1}), x_0).$ <p>По АΣ01 получаем:</p> $\Sigma(\Sigma(x_0, x_{n+1}), x_0) = \Sigma(x_0, x_0) = x_0.$
<p>С. 71.</p>	<p>С. 99.</p>
<p>Теорема 2.5. Для всіх ОПЮ виконується суворі рівність (еквівалентність): $\Sigma(x_1, \dots, x_k, x_0, x_{k+2}, \dots, x_n) = x_0$ при довільних k і n, що є натуральними числами.</p> <p>Для доказу цього твердження перетворюватимемо n-арну суму відповідно до рекурсивного визначення:</p> $\Sigma(x_1, \dots, x_k, x_0, x_{k+2}, \dots, x_n) = \Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_n\right) = \Sigma\left(\Sigma\left(\sum_{i=1}^{n-2} x_i\right), x_{n-1}\right) = \dots = \Sigma\left(\Sigma\left(\dots\left(\sum_{i=1}^k (x_k, x_0), x_{k+2}\right), \dots\right), x_n\right)$	<p>Теорема 2.5. Для всех ОПЮ выполняется строгое равенство (эквивалентность): $\Sigma(x_1, \dots, x_k, x_0, x_{k+2}, \dots, x_n) = x_0$ при произвольных k и n, являющихся натуральными числами.</p> <p>Для доказательства этого утверждения будем преобразовывать n-арную сумму в соответствии с рекурсивным определением:</p> $\Sigma(x_1, \dots, x_k, x_0, x_{k+2}, \dots, x_n) = \Sigma\left(\left(\sum_{i=1}^{n-1} x_i\right), x_n\right) = \Sigma\left(\Sigma\left(\sum_{i=1}^{n-2} x_i\right), x_{n-1}\right), x_n = \dots = \Sigma\left(\Sigma\left(\dots\left(\sum_{i=1}^k (x_k, x_0), x_{k+2}\right), \dots\right), x_n\right).$
<p>С. 71.</p>	<p>С. 99.</p>
<p>Оскільки тут змінюються два індекси k і n, то необхідно використовувати подвійну індукцію. Оскільки k зобов'язаний знаходитися в межах $0 < k < n$, то скористаємося зворотною індукцією по k:</p> <p>а) $k = 1, n = 3$:</p> $\Sigma x_0, x_2, x_3 = \Sigma(\Sigma(x_0, x_2), x_3) = \Sigma x_0, x_3 = x_0$ <p>Це витікає з асоціативного закону і Аксиоми В.1. Слід довести, що теорема справедлива при $k = 1$ і довільному n. Тому робимо наступний крок;</p> <p>б) $k = 1, n = m$, вважаючи, що твердження істинне при $n = m$, доведемо, що воно істинне і при $n = m + 1$:</p>	<p>Поскольку здесь изменяются два индекса k и n, то необходимо использовать двойную индукцию. Так как k обязан находиться в пределах $0 < k < n$, то воспользуемся обратной индукцией по k:</p> <p>а) $k = 1, n = 3$:</p> $\Sigma x_0, x_2, x_3 = \Sigma(\Sigma(x_0, x_2), x_3) = \Sigma x_0, x_3 = x_0.$ <p>Это следует из ассоциативного закона и АΣ01. Следует доказать, что теорема справедлива при $k = 1$ и произвольном n. Поэтому делаем следующий шаг;</p> <p>б) $k = 1, n = m$, считая, что утверждение истинно при $n = m$, докажем, что оно истинно и при $n = m+1$:</p>
<p>С. 71.</p>	<p>С. 100.</p>
$\Sigma x_0, x_2, x_3, \dots, x_{m+1} = \Sigma(\Sigma(x_0, \dots, x_m), x_{m+1}) = \Sigma x_0, x_{m+1} = x_0.$ <p>Далі переходимо до зворотної індукції по k.</p> <p>в) перевіряємо справедливість теореми при $k = n - 1$:</p> $\Sigma(x_1, \dots, x_0, x_n) = \Sigma(\Sigma(x_1, \dots, x_0), x_n) = \Sigma x_0, x_n = x_0$ <p>відповідно до закону нульової множини.</p>	$\Sigma x_0, x_2, x_3, \dots, x_{m+1} = \Sigma(\Sigma(x_0, \dots, x_m), x_{m+1}) = \Sigma x_0, x_{m+1} = x_0.$ <p>Далее переходим к обратной индукции по k.</p> <p>в) проверяем справедливость теоремы при $k = n - 1$:</p> $\Sigma(x_1, \dots, x_0, x_n) = \Sigma(\Sigma(x_1, \dots, x_0), x_n) = \Sigma x_0, x_n = x_0$ <p>в соответствии с законом нулевого множества.</p>

<p>С. 72.</p> <p>г) покажемо, що якщо теорема вірна для довільного k, $1 < k < n$, то вона вірна і для $k - 1$:</p> $\sum (x_1, \dots, x_{k-1}, x_0, x_{k+2}, \dots, x_n) = \sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_k) (x_0, x_{k+2}, \dots, x_n) = x_0 \quad (2.1)$ $\sum (x_1, \dots, x_{k-1}, x_0, x_{k+1}, \dots, x_n) = \sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_0) (x_{k+1}, x_{k+2}, \dots, x_n) =$ $\sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} (x_0, x_{k+1}, x_{k+2}, \dots, x_n) = \sum_{n-k-1 \text{ раз}} (\dots \sum_{i=1}^{k-1} (x_0, x_{k+2}, x_{k+3}, \dots, x_n) \quad (2.2)$	<p>С. 100.</p> <p>г) покажем, что если теорема верна для произвольного k, $1 < k < n$, то она верна и для $k - 1$:</p> $\sum (x_1, \dots, x_{k-1}, x_0, x_{k+2}, \dots, x_n) = \sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_k) (x_0, x_{k+2}, \dots, x_n) = x_0; \quad (2.1)$ $\sum (x_1, \dots, x_{k-1}, x_0, x_{k+1}, \dots, x_n) = \sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_0) (x_{k+1}, x_{k+2}, \dots, x_n) =$ $= \sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} (x_0, x_{k+1}, x_{k+2}, \dots, x_n) = \sum_{n-k-1 \text{ раз}} (\dots \sum_{i=1}^{k-1} (x_0, x_{k+2}, x_{k+3}, \dots, x_n)$
<p>С. 72.</p> <p>Суму (2.1) приведемо до вигляду (2.2), використовуючи закон нульової множини:</p> $\sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_k) (x_0, x_{k+2}, \dots, x_n) =$ $= \sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_0) (x_0, x_{k+2}, \dots, x_n) =$ $= \sum_{n-k-1 \text{ раз}} (\dots \sum_{i=1}^{k-1} (x_0, x_{k+2}, x_{k+3}, \dots, x_n)$	<p>С. 100.</p> <p>Сумму (2.1) приведем к виду (2.2), используя закон нулевого множества:</p> $\sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_k) (x_0, x_{k+2}, \dots, x_n) =$ $= \sum_{n-k \text{ раз}} (\dots \sum_{i=1}^{k-1} x_i, x_0) (x_{k+2}, \dots, x_n) =$ $= \sum_{n-k-1 \text{ раз}} (\dots \sum_{i=1}^{k-1} (x_0, x_{k+2}, x_{k+3}, \dots, x_n)$
<p>С. 72.</p> <p>Видно, що суми (2.1) і (2.2) набувають ідентичного вигляду, отже, твердження z справедливе, і, отже, істинна теорема в цілому.</p> <p>Слід зазначити, що комутативний закон для додавання ОПІО виконується для відношення структурної еквівалентності:</p> $(\Sigma(X, Y) R_{ST}(\Sigma(Y, X))).$ <p>Слюняєв неправильно переклав вислів «утверждение z» як «твердження z», не розуміючи букв. Смішний плагіат.</p>	<p>С. 100.</p> <p>Мы видим, что суммы (2.1) и (2.2) приобретают идентичный вид, следовательно, утверждение z справедливо, следовательно, истинна теорема в целом.</p> <p>Следует отметить, что коммутативный закон для суммирования ОПІО выполняется для отношения структурной эквивалентности:</p> $(\Sigma(X, Y) R_{ST}(\Sigma(Y, X))).$
<p>С. 72–73.</p> <p>Множина функцій інформаційного об'єкта.</p> <p>Введення поняття “множина функцій” дозволяє природним чином розділити ІО на два класи: активні ($\{F\} \neq \emptyset$) і пасивні ($\{F\} = \emptyset$).</p> <p>Активними ІО (АІО) називатимемо такі ІО, у яких множина функцій не порожня, тобто вони володіють власною поведінкою і можуть виконувати деякі активні дії.</p> <p>Пасивними ІО (ПІО) назвемо ІО, що не володіють власною поведінкою, пасивно беруть участь в реалізації деяких дій.</p>	<p>С. 107.</p> <p>2.2.4. Модель поведіння інформаційного об'єкта</p> <p>Введення поняття «модель поведінки» дозволяє естественным образом разделить ІО на два класса: активные ($BM \neq 0$) і пасивные ($BM = 0$).</p> <p>Активними ІО (АІО) будем называть такие ІО, у которых модель поведінки не пуста, т. е. они обладают собственным поведением и могут выполнять некоторые активные действия. Пассивными ІО (ПІО) назовем ІО, не обладающие собственным поведением, пассивно участвующие в реализации некоторых действий. В класс ПІО попадают определенные ранее структурные информационные объекты (СІО) и однопараметрические ІО (ОПІО).</p>
<p>С. 73.</p> <p>Припустимо, що взаємодія АІО здійснюється через прийом і передачу ПІО, при цьому в множині функцій АІО можуть породжуватися необхідні ПІО і передаватися іншому ІО. Цей інформаційний обмін можна трактувати як передачу повідомлень, обмін сигналами, зміну вхідних сигналів і тому подібне, що дозволяє розглядати систему “джерело – приймач” в більш широкому сенсі ніж парадигма “клієнт – сервер”.</p> <p>Традиційним засобом опису функціонування об'єктів в об'єктно-орієнтованих методах і системах є модель скінченого автомата. Як більш універ-</p>	<p>С. 107.</p> <p>Автор предполагает, что взаимодействие АІО осуществляется через прием и передачу ПІО, при этом в модели поведения АІО могут породжуватися необходимые ПІО и передаваться другому ІО. Этот информационный обмен можно трактовать как передачу сообщений, обмен сигналами, изменение входных сигналов и тому подобное, что позволяет рассматривать парадигму «источник - приемник» в широком смысле, вместо парадигмы «клиент — сервер».</p> <p>Традиционным средством описания поведения объектов в объектно-ориентированных методах и</p>

<p>сальний засіб визначення функціонування ІО пропонується використовувати апарат канонічних числень Е. Поста [106].</p> <p>Канонічним численням (КЧ) називатимемо четвірку вигляду</p> $(A, a, P, G), \quad (2.3)$ <p>де А – алфавіт числення, а – список аксіом КЧ, Р – алфавіт змінних, G – список правил виводу, кожне з яких має вигляд</p> $\frac{G_{11}P_{11}, G_{12}P_{12}, \dots, G_{1n_1}P_{1n_1}, G_{1n_1+1};}{G_{21}P_{21}, G_{22}P_{22}, \dots, G_{2n_2}P_{2n_2}, G_{2n_2+1};}$ $\frac{G_{m1}P_{m1}, G_{m2}P_{m2}, \dots, G_{mm_m}P_{mm_m}}{G_1P_1, G_2P_2, \dots, G_nP_n, G_{n+1}}$	<p>системах являється модель кінцевого автомата. В качестве более универсального средства определения поведения ИО автором предлагается использовать аппарат канонических исчислений Э. Поста [262].</p> <p>Каноническим исчислением (КИ) называется четверка вида [262]</p> $(A, a, P, G), \quad (2.3)$ <p>где А — алфавит исчисления, а — список аксиом КИ, Р - алфавит переменных, G — список правил вывода, каждое из которых имеет вид</p> $\frac{G_{11}P_{11}, G_{12}P_{12}, \dots, G_{1n_1}P_{1n_1}, G_{1n_1+1};}{G_{21}P_{21}, G_{22}P_{22}, \dots, G_{2n_2}P_{2n_2}, G_{2n_2+1};}$ $\frac{G_{m1}P_{m1}, G_{m2}P_{m2}, \dots, G_{mm_m}P_{mm_m}}{G_1P_1, G_2P_2, \dots, G_nP_n, G_{n+1}}$
<p>С. 74–75.</p>	<p>С. 107–108.</p>
<p>де G_{ij} ($i=1, \dots, m, j=1, \dots, n$) і $G(k=1, \dots, n+1)$ – деякі конкретні слова в алфавіті А, у тому числі і порожнє слово. Далі скористаємось визначеннями, введеними в [107–108]. Число t називається індексом схеми (2.4). Реалізуючим набором схеми (2.4) називатимемо вираз вигляду</p> $p^{(1)}, p^{(2)}, \dots, p^{(f)};$ P_1, P_2, \dots, P_f <p>де $p(1), p(2), \dots, p(f)$ – список (без повторень) всіх змінних, що входять в (2.4), а P_i – слово в алфавіті А, яке називається значенням змінної в даному реалізуючому наборі ($i=1, 2, \dots, f$). Якщо замість кожного входження кожної змінної в схему (2.4) підставити значення цієї змінної в R (R – це конкретний реалізуючий набір схеми (2.4)), то всі рядки схеми перетворяться на слова в алфавіті А. Так виходить реалізація продукційної схеми (2.4) реалізуючим набором R. Конструктивний об’єкт називатимемо реалізацією схеми (2.4), якщо він є реалізацією схеми яким-небудь реалізуючим набором. Слово Q називатимемо словом, що безпосередньо виводиться по схемі (2.4) із слів Q_1, Q_2, \dots, Q_m, якщо вираз</p> $Q_1, Q_2, \dots, Q_m \Rightarrow Q$ <p>є реалізацією схеми (2.4). Список слів називається выводом в КЧ, якщо кожне слово списку є аксіомою даного КЧ або безпосередньо виводиться по якій-небудь схемі із попередніх слів в даному списку. Довжиною виводу називатимемо число слів у выводі. Слово Р називатимемо таким, що виводиться в КЧ, якщо можна побудувати вивід в КЧ, останнім словом якого є Р.</p>	<p>где G_{ij} ($i=1, \dots, m, j=1, \dots, n$) и $G(k=1, \dots, n+1)$ - некоторые конкретные слова в алфавите А, в том числе и пустое слово. Далее воспользуемся определениями, введенными в [263], [264]. Число t называется индексом схемы (2.4). Реализующим набором схемы (2.4) называется выражение вида</p> $p^{(1)}, p^{(2)}, \dots, p^{(f)};$ $P_1, P_2, \dots, P_f,$ <p>где $p(1), p(2), \dots, p(f)$ - список (без повторений) всех переменных, входящих в (2.4), а P_j - слово в алфавите А, называемое значением переменной в данном реализующем наборе, ($i=1, 2, \dots, f$). Если вместо каждого вхождения каждой переменной в схему (2.4) подставить значение этой переменной в R (R - это конкретный реализующий набор схемы (2.4)), то все строки схемы превратятся в слова в алфавите А. Так получается реализация производящей схемы (2.4) реализующим набором R. Конструктивный объект называется реализацией схемы (2.4), если он является реализацией схемы каким-либо реализующим набором. Слово Q называется словом, непосредственно выводимым по схеме (2.4) из слов Q_1, Q_2, \dots, Q_m, если выражение</p> $\begin{matrix} Q_1 \\ Q_2 \\ \dots \\ \dots \\ Q_m \\ \dots \\ Q \end{matrix}$ <p>является реализацией схемы (2.4). Список слов называется выводом в КИ, если каждое слово списка является аксиомой данного КИ или непосредственно выводимо по какой-нибудь схеме из слов, предшествующих этому слову в рассматриваемом списке. Длинной вывода называется число слов в выводе. Слово Р называется выводимым в КИ, если можно построить вывод в КИ, последним словом которого является Р.</p>
<p>С. 75.</p>	<p>С. 108–109.</p>
<p>Поняття виводу і выводимості в КЧ дозволяють визначити еквівалентність числень і їх відношення з поняттям множини (2.4). Якщо алфавіт містить А, то воно називається численням над А. Два числення над А еквівалентні відносно А, якщо будь-яке слово в А виводиться в першому численні тоді і</p>	<p>Понятия вывода и выводимости в КИ позволяют определить эквивалентность исчислений и их отношения с понятием множества (2.4). Если алфавит содержит Л, то оно называется исчислением над А. Два исчисления над А эквивалентны относительно А, если любое слово в А выводимо в</p>

<p>тільки тоді, коли воно виводиться в другому. Пара $(A, T), (2.5)$ де T – канонічне числення над A, є поданням множини що виводяться в T слів алфавіту A. При цьому A називатимемо основним алфавітом подання, а доповнення до повного алфавіту числення T називатимемо допоміжним алфавітом, при цьому говорять, що T строго представляє множину слів, що виводяться в ньому. Множина M слів в A злічима, якщо для неї існує подання (2.5). Для формалізації функціональної моделі ІО введемо: $\{R\}$ – множина ПІО, що приймаються даним об'єктом; $\{T\}$ – множина передаваних даним об'єктом ПІО.</p>	<p>первом исчислении тогда и только тогда, когда оно выводимо во втором. Пара $(A, T), (2.5)$ где T - каноническое исчисление над A является представлением множества выводимых в T слов алфавита A. При этом L называется основным алфавитом представления, а дополнение до полного алфавита исчисления T называется вспомогательным алфавитом, при этом говорят, что T строго представляет множество выводимых в нем слов. Множество M слов в A перечислимо, если для него существует представление (2.5). Для формализации модели поведения ИО введем: $\{R\}$ - множество принимаемых данным объектом ПИО, $\{T\}$ - множество передаваемых данным объектом ПИО.</p>
<p>С. 75–76.</p>	<p>С. 109.</p>
<p>Вважаємо, що функціональна модель оперує з множинами $\{A\}$, $\{R\}$ і $\{T\}$. Оскільки для кожного атрибута A_i множина, на якій він визначений, – SA_i, може мати різну природу, то елементи цієї множини можна інтерпретувати досить широко: як програмні коди, виклики функцій операційної системи, графічні структури та ін. Також виділимо типи станів ІО у функціональній моделі: а) стани, в яких можливий прийом елементів множини $\{R\}$, надалі позначених як R_i; б) стани, в яких неможливий прийом R_i. Оскільки множина функцій повинна враховувати співвідношення атрибутів і зміст R_i, то необхідно ввести предикати, які утворюватимуть множину допустимих предикатів: $\{Pr\} = (Pr_1, Pr_2, \dots, Pr_\varphi).$</p>	<p>Полагаем, что ВМ оперирует с множествами $\{A\}$, $\{R\}$ и $\{T\}$. Поскольку для каждого атрибута A_i множество, на котором он определен - SA_i, - может иметь различную природу, то элементы этого множества можно интерпретировать весьма широко: как программные коды, вызовы функций операционной системы (например, Windows), графические структуры и т. д. Представляется необходимым введение понятия состояния ИО в ВМ и выделение двух типов состояний: а) состояния, в которых возможен прием элементов множества $\{R\}$у в дальнейшем обозначенных как R_i; б) состояния, в которых невозможен прием R_j. Так как в модели поведения необходимо учитывать соотношение атрибутов и содержание R_i, то необходимо введение предикатов, которые будут образовывать множество допустимых предикатов: $\{Pr\} = (Pr_1, Pr_2, \dots, Pr_\varphi).$</p>
<p>С. 76.</p>	<p>С. 109–110.</p>
<p>Для аналізу складних умов і співвідношень будуватимемо формули над предикатами в мові числення висловлювань і позначати їх $F(Pr)$ або просто F. Алфавіт числення КФМ визначимо таким чином: $A = (\{R\}, \{T\}, \{A\}, \{S\}, \{Pr\}, \&, \vee, (,), \neg, \rightarrow, \xi, \nabla, \emptyset)$, де $\{S\}$ – множина станів ІО, ξ – символ порожнього слова. До нього включаємо символи мови ІО для побудови формул F. Алфавіт змінних включатиме змінні $P = (p, q, f, hA)$, де p – послідовність вхідних ПІО, q – послідовність вихідних ПІО, f – послідовність формул з предикатами Pr в ІО, hA – список атрибутів ІО, для якого будується функціональна модель. Аксиому числення задамо як $A = (\emptyset \xi S_0 \xi hA(0) \xi \emptyset \xi \emptyset)$, де \emptyset означає порожній стан змінної, а під $hA(0)$ розуміється список вигляду $hA(0) = \langle \langle NA_1, SA_1, VA_1(0) \rangle, \langle NA_2, SA_2, VA_2(0) \rangle, \dots, \langle NA_n, SA_n, VA_n(0) \rangle \rangle$, де $VA_i(0)$ позначає значення i-го атрибуту у момент часу $t = 0$, тобто у момент початку функціонування ІО.</p>	<p>Для анализа сложных условий и соотношений будем строить формулы над предикатами в языке исчисления высказываний и обозначать их $F(Pr)$ или просто F. Алфавит исчисления Квм определим следующим образом: $A = (\{R\}, \{T\}, \{A\}, \{S\}, \{Pr\}, \&, \vee, (,), \neg, \rightarrow, \#, \nabla, \emptyset)$. В него включаем символы языка ИО для построения формул F. Алфавит переменных будет включать переменные $P = (p, q, f, hA)$, где p – последовательность входящих ПИО, q - последовательность выходящих ПИО, f – последовательность формул с предикатами Pr в ИО, hA - список атрибутов ИО, для которого строится ВМ. Аксиому исчисления зададим как $A = (\emptyset \# S_0 \# hA(0) \# \emptyset \# \emptyset)$, где \emptyset означает пустое состояние переменной, под $hA(0)$ понимается список вида $\langle NA_1, SA_1, VA_1(0) \rangle$; $\langle NA_2, SA_2, VA_2(0) \rangle$; ... $\langle NA_n, SA_n, VA_n(0) \rangle$, где $Vai(0)$ обозначает значение i-го атрибута в момент времени $t = 0$, т. е. в момент начала функ-</p>

		ціонирования ИО.
С. 77.		С. 110.
<p>Правила виводу для числення КФМ будувати- мемо як схеми правил, оскільки в конкретній функ- ціональній моделі виходитиме різна кількість пра- вил виводу, які мають вигляд, що задовольняє за- пропонованим схемам:</p> <p>Схема 1: $R_i p \# S_0 \# hA(0) \# q \# f \Rightarrow p \# S_i \# hA(R_i) \# q, T_i \# f, F_i$; (2.6)</p> <p>Схема 2: $R_i p \# S_0 \# hA(0) \# q \# f \Rightarrow \forall p \# S_i \# hA(R_i) \# q, T_i \# f, F_i$. (2.7)</p>	<p>Правила вывода для исчисления Квм будем строить как схемы правил, поскольку в конкретной ВМ будет получаться различное количество пра- вил вывода, имеющих вид, удовлетворяющий предлагаемым схемам:</p> <p>1) $\frac{R_i p \# S_0 \# hA(0) \# q \# f}{p \# S_i \# hA(R_i) \# q, T_i \# f, F_i}$;</p> <p>2) $\frac{R_i p \# S_0 \# hA(0) \# q \# f}{\forall p \# S_i \# hA(R_i) \# q, T_i \# f, F_i}$;</p>	
С. 77.		С. 111.
<p>Схеми (2.6) і (2.7) задають правила, що виво- дять зі стану S0 в стани типу а) і б) відповідно. Для позначення неможливості обробки вхідної послі- довності Ri використовується службовий символ □. У цих схемах породжується вихідний ПІО Ti, і фор- мула Fi, при цьому ми допускаємо можливість зав- дання Ti = ∅ і Fi = ∅, що дозволяє уникнути зайвих схем виводу.</p>	<p>Схемы 1 и 2 задают правила, выводящие из со- стояния Sq в состояние типа а и б соответственно. Для обозначения невозможности обработки вход- ной последовательности Rj используется служеб- ный символ V. В этих схемах порождается выхо- дящий ПИО Ti и формула Fi, при этом мы допуска- ем возможность задания Ti = ∅ и Fi = ∅, что позво- ляет избежать лишних схем вывода.</p>	
С. 77.		С. 110.
<p>Схема 3: $R_i p \# S_i \# hA \# q \# f \Rightarrow p \# S_j \# hA(R_i) \# q, T_j \# f, F_j$; (2.8)</p> <p>Схема 4: $R_i p \# S_i \# hA \# q \# f, F_i \Rightarrow p \# S_j \# hA(R_i) \# q, T_j \# f, F_j$. (2.9)</p>	<p>3) $\frac{R_i p \# S_i \# hA \# q \# f}{p \# S_j \# hA(R_i) \# q, T_j \# f, F_j}$;</p> <p>4) $\frac{R_i p \# S_i \# hA \# q \# f, F_i}{p \# S_j \# hA(R_i) \# q, T_j \# f, F_j}$;</p>	
С. 77.		С. 111.
<p>Схеми (2.8) і (2.9) визначають переходи із станів типу а) в стани типу а), з аналізом істинності Fi або без аналізу. Допускаємо також, що в Fi може бути задана формула (–Fi), тобто перевіряється істинність заперечення деякої формули. Таке роз- ширення допустиме, оскільки у численні висловлю- вавь істинність або помилковість будь-якого висло- ву може бути точно встановлена. Оброблена фор- мула Fi виключається з подальшого процесу виво- ду. При циклічній поведінці ІО необхідна формула може знову породжуватися схемами (2.8).</p>	<p>Схемы 3 и 4 определяют переходы из состоя- ний типа а в состояния типа а, с анализом истин- ности Fi или без анализа. Допускаем также, что в Fi может быть задана формула (–Fj), т. е. проверя- ется истинность отрицания некоторой формулы. Такое расширение допустимо, так как в ИВ истин- ность или ложность любого высказывания может быть точно установлена. Обработанная формула Fi исключается из дальнейшего процесса вывода. При циклическом поведении ИО необходимая формула может снова породжаться схемами 3.</p>	
С. 78.		С. 110.
<p>Схема 5: $R_i p \# S_i \# hA \# q \# f \Rightarrow \forall p \# S_j \# hA(R_i) \# q, T_j \# f, F_j$; (2.10)</p> <p>Схема 6: $R_i p \# S_i \# hA \# q \# f, F_i \Rightarrow \forall p \# S_j \# hA(R_i) \# q \# f$. (2.11)</p>	<p>5) $\frac{R_i p \# S_i \# hA \# q \# f}{\forall p \# S_j \# hA(R_i) \# q, T_j \# f, F_j}$;</p> <p>6) $\frac{R_i p \# S_i \# hA \# q \# f, F_i}{\forall p \# S_j \# hA(R_i) \# q \# f}$;</p>	
С. 78.		С. 111.
<p>Схема (2.10) задає перехід із стану типу а) в стан типу б) без аналізу F, а схема (2.11) – з аналі- зом F.</p>	<p>Схема 5 задает переход из состояния типа а в состояние типа б без анализа F, а схема 6 — с анализом F.</p>	
С. 78.		С. 110–111.
<p>Схема 7: $\forall p \# S_i \# hA \# q \# f \Rightarrow p \# S_j \# hA(R_i) \# q, T_j \# f, F_j$; (2.12)</p> <p>Схема 8: $\forall p \# S_i \# hA \# q \# f, F_i \Rightarrow p \# S_j \# hA(S_i) \# q, T_j \# f$; (2.13)</p> <p>Схема 9: $\forall p \# S_i \# hA \# q \# f \Rightarrow \forall p \# S_j \# hA(S_i) \# q, T_j \# f, F_j$; (2.14)</p> <p>Схема 10: $\forall p \# S_i \# hA \# q \# f, F_i \Rightarrow \forall p \# S_j \# hA(S_i) \# q, T_j \# f$. (2.15)</p>	<p>7) $\frac{\forall p \# S_i \# hA \# q \# f}{p \# S_j \# hA(R_i) \# q, T_j \# f, F_j}$;</p> <p>8) $\frac{\forall p \# S_i \# hA \# q \# f, F_i}{p \# S_j \# hA(S_i) \# q, T_j \# f}$;</p> <p>9) $\frac{\forall p \# S_i \# hA \# q \# f}{\forall p \# S_j \# hA(S_i) \# q, T_j \# f, F_j}$;</p> <p>10) $\frac{\forall p \# S_i \# hA \# q \# f, F_i}{\forall p \# S_j \# hA(S_i) \# q, T_j \# f}$.</p>	

<p>С. 78.</p> <p>Схеми (2.12), (2.13) визначають переходи із станів типу б) в стани типу а) , а схеми (2.14) і (2.15) із станів типу б) в стани типу б) . У цих схемах правил закладається можливість повернення в стан S0 і зупинки при переході в такий стан Sj з якого немає можливості подальшого виводу.</p> <p>У схемах правил виводу (2.6) – (2.15) задаються в загальному вигляді функціональні перетворення h A (Si), які можна визначити як перетворення над значеннями атрибутів: $V_{ai} := f(k) (VA\phi_1, \dots, VA\phi_k)$, де k – арність функціонального символу, VAfi – i-й аргумент функції f(k), узятий із списку значень атрибутів.</p>	<p>С. 111.</p> <p>Схеми 7, 8 определяют переходы из состояний типа б в состояния типа а, а схемы 9 и 10 из состояний типа б в состояния типа б. В этих схемах правил закладывается возможность возврата в состояние S0 и останова при переходе в такое состояние Sj, из которого нет возможности дальнейшего вывода.</p> <p>В схемах правил вывода 1-10 задаются в общем виде функциональные преобразования hA(Sj), которые можно определить как преобразования над значениями атрибутов: $V_{Ai} := f(k)(VA\phi_1, \dots, VA\phi_k)$, где k - арность функционального символа, VAfi — j-й аргумент функции f(k), взятый из списка значений атрибутов, причем $SA_{i1} \equiv SA_{\phi 1} \equiv SA_{\phi 2} \equiv \dots \equiv SA_{\phi k}$, т. е. значения VAφ1, ..., VAφk должны быть определены на одном и том же множестве.</p>
<p>С. 79–80.</p>	<p>С. 138.</p>
<p>Комбінована модель інтелектуального агента ІКСА. На підставі аналізу характеристик та недоліків відомих моделей ІА, пропонується визначати ІА як структуру вигляду</p> $IA = \langle N_{IA}, SA, V_{IA}, M_{VB}, VO \rangle,$ <p>де NIA – ім'я інтелектуального агента; SA – структура атрибутів, яка визначається аналогічно структурі атрибутів для інформаційних об'єктів (ІО) [111]; VIA = {IA} – множина вкладених ІА; MVB – механізм вибору моделі функціонування, VO = {O} – множина інформаційних об'єктів, що реалізують сценарії роботи ІА.</p> <p>Інтелектуальний агент на підставі критеріїв вибору моделі функціонування, закладених в MVB, приймає рішення про реалізацію в даний момент часу деякого сценарію роботи і ініціалізує відповідний ІО.</p>	<p>2.3.2. Обобщенная модель интеллектуального агента КИНС ППР</p> <p>На основании изложенных выше соображений о характеристиках известных моделей ИА, автор предлагает определить ИА как структуру вида</p> $IA = \langle N_{IA}, SA, V_{IA}, M_{VB}, VO \rangle,$ <p>где NIA - имя интеллектуального агента, удовлетворяющее соглашению об именах; SA - структура атрибутов, определяемая аналогично структуре атрибутов для ИО; VIA = {IA} - множество ИЛ, вложенных в данный или образующих следующий уровень иерархии ИА; MVB - механизм выбора поведения, VO= {O} - множество информационных объектов, реализующих сценарии поведения ИА. ИА на основании критериев выбора поведения, заложенных в MVB, принимает решение о реализации в данный момент времени некоторого сценария поведения и инициализирует соответствующий ИО.</p>
<p>С. 80.</p>	<p>С. 138.</p>
<p>Інтелектуальних агентів, які мають лише ім'я (IA = <NIA, ∅, ∅, ∅, ∅ >), можна назвати номінальними, за аналогією з номінальними ІО. За своїм змістом номінальні ІА нічим не відрізняються від номінальних ІО. Подібна ситуація відбувається і з класом параметричних ІА (IA = <NIA, SA, ∅, ∅, ∅ >). За ознакою наявності механізмів поведінки ІА поділяються на два класи: активні (MVB ≠ ∅ & VO ≠ ∅) і пасивні (MVB = ∅ & VO = ∅). У класі пасивних ІА виділяють підклас пасивних агентів – оболонки, які використовуються для подання і зв'язку об'єктів у системі (IA = <NIA, ∅, VIA, ∅, ∅ >). Власне пасивні ІА (IA = <NIA, SA, VIA, ∅, ∅ >) мають атрибути, які носять характер констант і дозволяють створити деякий проміжний рівень ієрархії. У класі активних ІА виникає підклас невизначених ІА (IA = <NIA, ∅, ∅, MVB, VO > ∪ <NIA, ∅, VIA, MVB, VO >), який передбачає маніпуляції з внутрішньою структурою атрибутів.</p>	<p>Інтелектуальных агентов, обладающих только именем (IA = <NIA, ∅, ∅, ∅, ∅ >), можно назвать номинальными, по аналогии с номинальными ИО (см. раздел 2.2). По своему содержанию номинальные ИА ничем не отличаются от номинальных ИО. Подобная ситуация возникает и с классом параметрических ИА (IA = <NIA, SA, ∅, ∅, ∅ >). По признаку наличия механизмов поведения ИА делятся на два класса: активные (MVB ≠ ∅ & VO ≠ ∅) и пассивные (MVB = ∅ & VO = ∅). В классе пассивных ИА выделяют подкласс пассивных агентов - оболочек, которые используются для представления и связи объектов в системе (IA = <NIA, ∅, VIA, ∅, ∅ >). Собственно пассивные ИА (IA = <NIA, SA, VIA, ∅, ∅ >) имеют атрибуты, носящие характер констант, и позволяют создать некоторый промежуточный уровень иерархии. В классе активных ИА возникает подкласс неопределенных ИА (IA = <NIA, ∅, ∅, MVB, VO > ∪ <NIA, ∅, VIA, MVB, VO >), который является внутренне противоречивым, так как предполагает манипуляции с пустой структурой атрибутов.</p>
<p>С. 80–81.</p>	<p>С. 138–139.</p>
<p>Класифікація ІА представлена в табл. 2.4 і 2.5. Припустимо, що структура атрибутів SA є загальним представленням простору знань ІА, тобто можна розглядати цю структуру на концептуальному рівні. У більшості відомих робіт проводиться,</p>	<p>Классификация ИА представлена в табл. 2.4 и на рис. 2.3. Автор предполагает, что структура атрибутов SA является обобщенным представлением пространства знаний ИА, то есть рассматривает эту структуру на концептуальном уровне. В боль-</p>

<p>функціональне розділення простору знань ІА (знання про середовище, знання про себе, знання про інших агентів та ін.). Тут “-” відповідає порожньому стану змінної (\emptyset), “+” позначає наявність даної змінної в моделі ІА.</p>	<p>шинстве известных работ проводится функциональное разделение пространства знаний ИА (знания о среде, знания о себе, знания о других агентах и т.п.). Здесь 0 соответствует пустому состоянию переменной (\emptyset), 1 обозначает наличие данной переменной в модели ИЛ.</p>																																																																																																																												
<p>С. 81.</p>	<p>С. 139.</p>																																																																																																																												
<p style="text-align: center;">Таблица 2.4</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th rowspan="2">Клас інтелектуальних агентів</th> <th colspan="5">Змінні моделі інтелектуального агента</th> </tr> <tr> <th>N_{IA}</th> <th>S_A</th> <th>V_{IA}</th> <th>M_{VZ}</th> <th>V_O</th> </tr> </thead> <tbody> <tr> <td>Порожні</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Номинальні</td> <td>+</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Параметричні</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Пасивні – оболонки</td> <td>+</td> <td>-</td> <td>+</td> <td>-</td> <td>-</td> </tr> <tr> <td>Пасивні</td> <td>+</td> <td>+</td> <td>+</td> <td>-</td> <td>-</td> </tr> <tr> <td>Активні</td> <td>+</td> <td>-</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>Активні невизначені</td> <td>+</td> <td>+</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>Активні термінальні</td> <td>+</td> <td>+</td> <td>-</td> <td>+</td> <td>+</td> </tr> <tr> <td>Активні</td> <td>+</td> <td>+</td> <td>+</td> <td>+</td> <td>+</td> </tr> </tbody> </table> <p style="color: red; font-weight: bold;">Слюняев скопировал чужую таблицу, переставивши последний столбчик леворуч и заменивши 0 та 1 на «-» та «+» відповідно. Плагіат.</p>	Клас інтелектуальних агентів	Змінні моделі інтелектуального агента					N_{IA}	S_A	V_{IA}	M_{VZ}	V_O	Порожні	-	-	-	-	-	Номинальні	+	-	-	-	-	Параметричні	+	+	-	-	-	Пасивні – оболонки	+	-	+	-	-	Пасивні	+	+	+	-	-	Активні	+	-	-	+	+	Активні невизначені	+	+	-	+	+	Активні термінальні	+	+	-	+	+	Активні	+	+	+	+	+	<p style="text-align: center;">Таблица 2.4</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="5">Переменные модели интеллектуального агента</th> <th rowspan="2">Класс интеллектуальных агентов</th> </tr> <tr> <th>N_{IA}</th> <th>S_A</th> <th>V_{IA}</th> <th>M_{VZ}</th> <th>V_O</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Пустые</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Номинальные</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>Параметрические</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>Пассивные – оболочки</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>Пассивные</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>Активные неопределенные</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>Активные терминальные</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>Активные</td> </tr> </tbody> </table>	Переменные модели интеллектуального агента					Класс интеллектуальных агентов	N_{IA}	S_A	V_{IA}	M_{VZ}	V_O	0	0	0	0	0	Пустые	1	0	0	0	0	Номинальные	1	1	0	0	0	Параметрические	1	0	1	0	0	Пассивные – оболочки	1	1	1	0	0	Пассивные	1	0	0	1	1	Активные неопределенные	1	0	1	1	1	Активные терминальные	1	1	0	1	1	Активные
Клас інтелектуальних агентів		Змінні моделі інтелектуального агента																																																																																																																											
	N_{IA}	S_A	V_{IA}	M_{VZ}	V_O																																																																																																																								
Порожні	-	-	-	-	-																																																																																																																								
Номинальні	+	-	-	-	-																																																																																																																								
Параметричні	+	+	-	-	-																																																																																																																								
Пасивні – оболонки	+	-	+	-	-																																																																																																																								
Пасивні	+	+	+	-	-																																																																																																																								
Активні	+	-	-	+	+																																																																																																																								
Активні невизначені	+	+	-	+	+																																																																																																																								
Активні термінальні	+	+	-	+	+																																																																																																																								
Активні	+	+	+	+	+																																																																																																																								
Переменные модели интеллектуального агента					Класс интеллектуальных агентов																																																																																																																								
N_{IA}	S_A	V_{IA}	M_{VZ}	V_O																																																																																																																									
0	0	0	0	0	Пустые																																																																																																																								
1	0	0	0	0	Номинальные																																																																																																																								
1	1	0	0	0	Параметрические																																																																																																																								
1	0	1	0	0	Пассивные – оболочки																																																																																																																								
1	1	1	0	0	Пассивные																																																																																																																								
1	0	0	1	1	Активные неопределенные																																																																																																																								
1	0	1	1	1	Активные терминальные																																																																																																																								
1	1	0	1	1	Активные																																																																																																																								
<p>С. 81.</p>	<p>С. 140.</p>																																																																																																																												
<p style="text-align: center;">Таблица 2.5</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">ИНТЕЛЛЕКТУАЛЬНЫЕ АГЕНТЫ</th> </tr> <tr> <th colspan="2">ПАСИВНЫЕ <$N_{IA}, S_A, \{IA\}, 0, 0$></th> <th colspan="2">АКТИВНЫЕ <$N_{IA}, S_A, \{IA\}, M_{VZ}, 0$></th> </tr> </thead> <tbody> <tr> <td>ПАСИВНО-ОБОЛОЧКИ <$N_{IA}, 0, \{IA\}, 0, 0$></td> <td>ПАРАМЕТРИЧНЫЕ <$N_{IA}, S_A, 0, 0, 0$></td> <td>ТЕРМИНАЛЬНЫЕ <$N_{IA}, S_A, 0, M_{VZ}, \{0\}$></td> <td>АКТИВНО НЕВИЗНАЧЕНІ <$N_{IA}, 0, 0, M_{VZ}, \{0\}$></td> </tr> <tr> <td></td> <td>ПОРОЖНИЙ <$0, 0, 0, 0, 0$></td> <td></td> <td>U <$N_{IA}, 0, \{IA\}, M_{VZ}, \{0\}$></td> </tr> </tbody> </table> <p style="color: red; font-weight: bold;">Слюняев переробив чужую схему в таблицу. Нахабный плагіат.</p>	ИНТЕЛЛЕКТУАЛЬНЫЕ АГЕНТЫ				ПАСИВНЫЕ < $N_{IA}, S_A, \{IA\}, 0, 0$ >		АКТИВНЫЕ < $N_{IA}, S_A, \{IA\}, M_{VZ}, 0$ >		ПАСИВНО-ОБОЛОЧКИ < $N_{IA}, 0, \{IA\}, 0, 0$ >	ПАРАМЕТРИЧНЫЕ < $N_{IA}, S_A, 0, 0, 0$ >	ТЕРМИНАЛЬНЫЕ < $N_{IA}, S_A, 0, M_{VZ}, \{0\}$ >	АКТИВНО НЕВИЗНАЧЕНІ < $N_{IA}, 0, 0, M_{VZ}, \{0\}$ >		ПОРОЖНИЙ < $0, 0, 0, 0, 0$ >		U < $N_{IA}, 0, \{IA\}, M_{VZ}, \{0\}$ >	<p style="text-align: center;">Рис. 2.3. Классификация интеллектуальных агентов</p>																																																																																																												
ИНТЕЛЛЕКТУАЛЬНЫЕ АГЕНТЫ																																																																																																																													
ПАСИВНЫЕ < $N_{IA}, S_A, \{IA\}, 0, 0$ >		АКТИВНЫЕ < $N_{IA}, S_A, \{IA\}, M_{VZ}, 0$ >																																																																																																																											
ПАСИВНО-ОБОЛОЧКИ < $N_{IA}, 0, \{IA\}, 0, 0$ >	ПАРАМЕТРИЧНЫЕ < $N_{IA}, S_A, 0, 0, 0$ >	ТЕРМИНАЛЬНЫЕ < $N_{IA}, S_A, 0, M_{VZ}, \{0\}$ >	АКТИВНО НЕВИЗНАЧЕНІ < $N_{IA}, 0, 0, M_{VZ}, \{0\}$ >																																																																																																																										
	ПОРОЖНИЙ < $0, 0, 0, 0, 0$ >		U < $N_{IA}, 0, \{IA\}, M_{VZ}, \{0\}$ >																																																																																																																										
<p>С. 81–82.</p>	<p>С. 139.</p>																																																																																																																												
<p>Функціональне розділення не є універсальним, оскільки залежить від особливостей прикладної системи і пов'язане з конкретною науочною областю. Концептуальний же розгляд дозволяє створювати більш загальні моделі інтелектуальних агентів при необхідності специфікуючи і уточнюючи структуру простору знань. ІА, що відносяться до даного інтелектуального компонента, утворюють як би “ближню околицю” простору наочної області і визначають свою поведінку, використовуючи ті або інші механізми соціальної поведінки в цій локалізованій групі. ІА користується механізмом логічного виводу і засобами встановлення несуперечності, розташованими на інтелектуальних компонентах. Інтелектуальний компонент буде стійкішою стаціонарною структурою, “жорстко” локалізованою логічно і просторово. ІА мобільніші, вони дістають можливість рухатися в мережевому середовищі. Оскільки ІА використовують загальний механізм логічного виводу, то відпадає необхідність його дублювання в кожному агентіві, що повинно підвищити</p>	<p>Функциональное разделение не является универсальным, так как зависит от особенностей прикладной системы и связано с конкретной предметной областью. Концептуальное же рассмотрение позволяет создавать более общие модели интеллектуальных агентов при необходимости специфицируя и уточняя структуру пространства знаний. ИА, относящиеся к данному ИК, образуют как бы «ближнюю окрестность» пространства предметной области и определяют свое поведение, используя те или иные механизмы социального поведения в этой локализованной группе. ИА пользуется механизмом логического вывода и средствами установления непротиворечивости, располагающимися на ИК. ИК будет более устойчивой стационарной структурой, «жестко» локализованной логически и пространственно. ИА более мобильны, они получают возможность перемещаться в сетевой среде. Поскольку ИА используют общий механизм логического вывода, то отпадает необходимость его удвоения в каждом агенте, что должно повысить</p>																																																																																																																												

швидкодію і скоротити витрати пам'яті.	быстродействие и сократит затраты памяти.
С. 82.	С. 141.
<p>Визначимо інформаційний простір IAi як сукупність інтелектуальних агентів, що оточують IAi і взаємодіють з ним; сукупність IO, що оточують IAi і взаємодіють з ним, і множину атрибутів, які необхідні IAi для оцінки стану навколишнього середовища</p> $V_{IA_i} = (AR_{IA_i}^i, AR_{IO}^i),$ $AR_{IA_i}^i = (N_{IA_j}, A_{IA_j}^{\xi}, \dots, A_{IA_j}^{\psi}, N_{IA_1}, A_{IA_1}^{\xi}, \dots, A_{IA_1}^{\psi}),$ $AR_{IO}^i = (N_{IO_j}, A_{IO_j}^{\xi}, \dots, A_{IO_j}^{\psi}, N_{IO_1}, A_{IO_1}^{\xi}, \dots, A_{IO_1}^{\psi}).$	<p>Определим информационное пространство IA; как совокупность интеллектуальных агентов, окружающих IAi и «интересующих» его, совокупность IO, окружающих IAi и «интересующих» его, и множеств атрибутов, которые необходимы IA(Для оценки состояния окружающей его среды</p> $V_{IA_i} = (AR_{IA_i}^i, AR_{IO}^i),$ $AR_{IA_i}^i = (NIA_j, A_{IA_j}^{\xi}, \dots, A_{IA_j}^{\psi}, \dots, NIA_1, A_{IA_1}^{\xi}, \dots, A_{IA_1}^{\psi}),$ $AR_{IO}^i = (NIO_j, A_{IO_j}^{\xi}, \dots, A_{IO_j}^{\psi}, \dots, NIO_1, A_{IO_1}^{\xi}, \dots, A_{IO_1}^{\psi}).$
С. 82–83.	С. 141.
<p>Станом інформаційного простору IAi назовемо сукупність значень ARⁱ_{IA}, ARⁱ_{IO} у момент часу t:</p> $SV_{IA_i} = (\langle A_{IA_j}^{\xi} \rangle, \dots, \langle A_{IA_j}^{\psi} \rangle, \dots, \langle A_{IA_1}^{\xi} \rangle, \dots, \langle A_{IA_1}^{\psi} \rangle, \dots, \langle A_{IO_j}^{\xi} \rangle, \dots, \langle A_{IO_j}^{\psi} \rangle, \dots, \langle A_{IO_1}^{\xi} \rangle, \dots, \langle A_{IO_1}^{\psi} \rangle),$ <p>де $\langle A_{...} \rangle$ – значення атрибута у момент часу t з погляду даного інтелектуального агента. Інформацію про стан інформаційного простору IAi отримує не одночасно і не одночасно від усіх точок цього простору, оскільки виникає затримка при обміні інформацією між агентами.</p> <p>Інформаційний простір агента може формуватися двома способами:</p> <p>а) статично – ARⁱ_{IA}, iARⁱ_{IO} визначені на стадії проектування IA і в процесі роботи ІКСА не міняються;</p> <p>б) динамічно –ARⁱ_{IA}, ARⁱ_{IO} можуть змінюватися в процесі функціонування ІКСА .</p>	<p>Состоянием информационного пространства IA назовем совокупность значений ARⁱ_{IA}, ARⁱ_{IO} в момент времени t:</p> $SV_{IA_i} = (\langle A_{IA_j}^{\xi} \rangle, \dots, \langle A_{IA_j}^{\psi} \rangle, \dots, \langle A_{IA_1}^{\xi} \rangle, \dots, \langle A_{IA_1}^{\psi} \rangle, \dots, \langle A_{IO_j}^{\xi} \rangle, \dots, \langle A_{IO_j}^{\psi} \rangle, \dots, \langle A_{IO_1}^{\xi} \rangle, \dots, \langle A_{IO_1}^{\psi} \rangle)$ <p>где $\langle A_{...} \rangle$ есть значение атрибута в момент времени t с точки зрения данного интеллектуального агента. Информацию о состоянии информационного пространства IAi получит не одночасно и не одновременно от всех точек этого пространства, так как возникнет задержка при обмене информацией между агентами.</p> <p>Информационное пространство агента может формироваться двумя способами:</p> <p>а) статически - ARⁱ_{IA}, ARⁱ_{IO} определены на стадии проектирования IA и в процессе работы КИНС ППР не меняются;</p> <p>б) динамически - ARⁱ_{IA}, ARⁱ_{IO} могут изменяться в процессе функционирования КИНС ППР.</p>
С. 83.	С. 141.
<p>Отже, модель інформаційного простору IAi визначається як</p> $MIS_{IA_i} = (V_{IA_i}(t), SV_{IA_i}(t), FV_{IA_i}(t+1)),$ <p>де FV – функція формування інформаційного простору.</p> <p>Модель механізму вибору поведінки IA приймає наступний вигляд:</p> $MVB = (MIS, MG, MSR, MA),$ <p>де MG – модель цілевизначення, MSR – модель пошуку рішення (пошуку шляхів досягнення мети), MA – модель активних дій, тобто механізм активізації IO, що впливають на середовище.</p>	<p>Следовательно, модель информационного пространства IAi определится как</p> $MIS_{IA_i} = (V_{IA_i}(t), SV_{IA_i}(t), FV_{IA_i}(t+1))$ <p>где FV - функция формирования информационного пространства.</p> <p>Модель механизма выбора поведения IA принимает следующий вид:</p> $MVB = (MIS, MG, MSR, MA),$ <p>где MG - модель целеполагания, MSR - модель поиска решения (поиска путей достижения цели), MA - модель активных действий, то есть механизм активизации IO, воздействующих на среду.</p>
С. 84.	С. 142.
<p>Для даного IA модель цілевизначення будується таким чином</p> $MG_{IA_i} = (SS_{IA_i}, FSS_{IA_i}, GS_{IA_i}, G_{IA_i}^{top}, G_{IA_i}^{down}, FG_{IA_i}^D, FG_{IA_i}^S, FAG_{IA_i}, SMA_{IA_i}(t))$ <p>Далі нижні індекси опустимо там, де це не викликає різночитань. Тут SS – множина стратегій, що розуміються як методи вибору цілей SS = (Si i=1, ..., n), FSS – функція вибору стратегії; GS – множина статичних цілей, G^{top} – множина цілей, що отримуються даним IA від агентів більш високого рівня ієрархії, G^{down} – множина цілей, які можуть бути передані IA нижніх рівнів; FGD – функція формування динамічних цілей, FGS – функція вибору статичних цілей; FAG – функція вибору активних цілей, тобто цілей, прийнятих до реалізації; SMA – стан навколишнього мультиагентного оточення.</p>	<p>Для данного IA модель целеполагания строится следующим образом:</p> $MG_{IA_i} = (SS_{IA_i}, FSS_{IA_i}, GS_{IA_i}, G_{IA_i}^{top}, G_{IA_i}^{down}, FG_{IA_i}^D, FG_{IA_i}^S, FAG_{IA_i}, SMA_{IA_i}(t))$ <p>Далее нижние индексы опустим там, где это не вызывает разночтений. Здесь SS - множество стратегий, понимаемых как методы выбора целей, SS = (Sij= 1, ...,n), FSS - функция выбора стратегии; GS - множество статических целей, G^{top} – множество целей, получаемых данным IA от агентов более высокого уровня иерархии, G^{down} - множество целей, которые могут быть переданы IA нижежащих уровней; FG^o - функция формирования динамических целей, FGS - функция выбора статических целей; FAG - функция выбора активных целей, т. е. целей, принятых к реализации; SMA - со-</p>

		стояние окружающего мультиагентного мира.
С. 84.	<p>На відміну від існуючих агентних моделей стан МА-оточення більш доцільно визначати з урахуванням динаміки його розвитку, враховуючи як минулу історію, так і очікуване майбутнє. Стан МА-оточення розглядається з позиції даного інтелектуального агента IA_i, тому</p> $SMA_{IA_i}(t) = (Pa_{IA_i}(t), Rt_{IA_i}(t), Fu_{IA_i}(t)).$ <p>Минуле МА-оточення є $Pa_{IA_i}(t) = \bigcup_0^t (V_{IA_i}(t-1), SV_{IA_i}(t-1))$, тобто об'єднання інформаційного простору і його станів за сукупністю попередніх моментів часу.</p> <p>Поточний стан МА-оточення $Rt_{IA_i}(t) = (V_{IA_i}(t), SV_{IA_i}(t))$.</p>	С. 142.
		<p>В отличие от агентных моделей, предложенных в [209], [211], [302]-[304] автор определяет состояние МА-мира с учетом динамики его развития, учитывая как прошлую историю, так и ожидаемое будущее. Состояние МА-мира рассматривается с позиции данного интеллектуального агента IA_i; поэтому</p> $SMA_{IA_i}(t) = (Pa_{IA_i}(t), Rt_{IA_i}(t), Fu_{IA_i}(t)).$ <p>Прошлое, «история» МА-мира есть</p> $Pa_{IA_i}(t) = \bigcup_0^t (V_{IA_i}(t-1), SV_{IA_i}(t-1)),$ <p>то есть объединение информационного пространства и его состояний по совокупности предшествующих моментов времени.</p> <p>«Настоящее» МА-мира (текущее состояние) $Rt_{IA_i}(t) = (V_{IA_i}(t), SV_{IA_i}(t))$.</p>
С. 85.	<p>Передбачуваний майбутній стан $Fu_{IA_i}(t) = (V_{IA_i}(t+1), SV_{IA_i}(t+1))$ – це оцінка інформаційного простору і його стану, виконана у момент часу $t-1$ тобто на попередньому кроці функціонування IA_i. Для здійснення цієї оцінки необхідна функція прогнозу майбутнього МА – оточення $FP(Rt_{IA_i}(t), MA)$ результатом дії якої і буде $Fu_{IA_i}(t)$.</p> <p>Функція вибору стратегії визначає поточну стратегію залежно від попередньої стратегії, стану МА-оточення, множини активних на даний момент цілей. Таким чином $FSS: s(t) \times SMA \times GA \rightarrow s(t)$.</p> <p>Якщо позначити статичні цілі як gs, цілі отримувани від вищих агентів як gt, цілі передавані нижчим рівням, як gd то відповідні множини запишуться у вигляді: $GS_{IA_i} = \{gs^i i = 1, \dots, m\}$, $G_{IA_i}^{top} = \{gt^i i = 1, \dots, l\}$, $G_{IA_i}^{down} = \{gd^i i = 1, \dots, k\}$.</p>	С. 142–143.
		<p>Предполагаемое будущее состояние $Fu_{IA_i}(t) = (V_{IA_i}(t), SV_{IA_i}(t))$ – это оценка информационного пространства и его состояния, выполненная в момент времени $t-1$ т. е. на предыдущем шаге функционирования IA_i. Для осуществления этой оценки необходима функция предсказания будущего МА-мира $FP(Rt_{IA_i}(t), MA)$ результатом действия которой и будет $Fu_{IA_i}(t)$.</p> <p>Функция выбора стратегии определяет текущую стратегию в зависимости от предыдущей стратегии, состояния МА-мира, множества активных (реализуемых) на данный момент целей $GA_{IA_i} = \{GA_{IA_i}^j j = 1, \dots, n\}$. Таким образом, $FSS: s(t) \times SMA \times GA \rightarrow s(t)$</p> <p>Если мы обозначим статические цели как gs, цели получаемые от вышестоящих агентов как gt цели передаваемые нижележащим уровням как gd то соответствующие множества запишутся в виде:</p> $GS_{IA_i} = \{gs^i i = 1, \dots, m\},$ $G_{IA_i}^{TOP} = \{gt^i i = 1, \dots, l\},$ $G_{IA_i}^{DOWN} = \{gd^i i = 1, \dots, k\}.$
С. 85.		С. 143.
	<p>Функція формування динамічних цілей визначається функціональним перетворенням h^D IA_i над станом МА-оточення, поточною стратегією, множинами $G_{AVT}(D)(t)$, $GA(t)$ і сукупністю формул в мові логіки першого порядку над елементами МА-оточення: $FG_{IA_i}^D = h^D(SMA_{IA_i}(t), s^i(t), G^{AVT(D)}(t), GA(t), U)$, де $U = \{U^j(SMA(t)) j = 1, \dots, k\}$. Результатом роботи $h_{IA_i}^D$ буде множина $G_{IA_i}^{AVT(D)}(t+1)$.</p>	<p>Функция формирования динамических целей определяется функциональным преобразованием h^{DI} над состоянием МА-мира, текущей стратегией, множествами $GAFT(D)(t)$, $GA(t)$ и совокупностью формул в языке логики первого порядка над элементами МА-мира:</p> $FG_{IA_i}^D = h^D(SMA_{IA_i}(t), s^i(t), G^{AVT(D)}(t), GA(t), U),$ <p>где $U = \{U^j(SMA(t)) j = 1, \dots, k\}$. Результатом работы $h_{IA_i}^D$ будет множество $G_{IA_i}^{AVT(D)}(t+1)$.</p>
С. 85–86.		С. 143.
	<p>Функція вибору статичних цілей визначається функціональним перетворенням h^S IA_i над станом МА-оточення, поточною стратегією, множиною сформованих статичних цілей на даний момент – $GAVT(D)(t)$, множиною активних цілей, прийнятих до виконання і сукупністю формул на мові логіки першого порядку над елементами МА-оточення:</p>	<p>Функция выбора статических целей определяется функциональным преобразованием h^{SI} над состоянием МА-мира, текущей стратегией, множеством сформированных статических целей на текущий момент – $GAVT(D)(t)$ множеством активных целей, принятых к исполнению и совокупностью формул в языке логики первого порядка над элементами МА-мира:</p>

$FG_{IA_k}^S = h^S(SMA_{IA_k}(t), s^i(t), G^{AVT(S)}(t), GA(t), W),$ <p>де $W = \{W^j(SMA(t)) j=1, \dots, k\}$. Результатом роботи $h_{IA_k}^S$ буде множина $G_{IA_k}^{AVT(S)}(t+1)$.</p>	$FG_{IA_k}^S = h^S(SMA_{IA_k}(t), s^i(t), G^{AVT(S)}(t), GA(t), W),$ <p>де $W = \{W^j(SMA(t)) j=1, \dots, k\}$. Результатом роботи $h_{IA_k}^S$ буде множина $G_{IA_k}^{AVT(S)}(t+1)$.</p>
<p>С. 86.</p>	<p>С. 164.</p>
<p>2.3. Розробка моделі процесу пошуку рішення інтелектуальним агентом інформаційно-керуючої системи аеропорту</p> <p>В ході аналізу відомих моделей ІА встановлено той факт, що в даних моделях застосовуються різні способи пошуку рішення. Під пошуком рішення слід розуміти знаходження шляху досягнення мети або цілей даним ІА в поточному стані МА-оточення. Оскільки різні структурні підрозділи аеропорту володіють своєю специфікою в т.ч. і при прийнятті рішень, то навряд чи є можливим застосування деякого універсального методу пошуку рішення для всіх підсистем ІКАС.</p> <p>Більш доцільним є розробка на базі узагальненої моделі ІА “бібліотеки” моделей, які дозволять створювати різні конкретні моделі ІА.</p> <p>З погляду реалізації дій відомі моделі слід розбити на три класи:</p> <p>1) з визначеною кінцевою множиною елементарних дій [112–117];</p>	<p>2.3.5. Модель поиска решения интеллектуального агента</p> <p>В ходе анализа известных моделей ИА установлен тот факт, что в данных моделях применяются различные способы поиска решения. Под поиском решения автор понимает нахождение пути достижения цели или целей данным ИА в текущем состоянии МА-мира. Поскольку разнообразие возможных сфер применения КИНС ППР и МАС весьма велико, то вряд ли представляется возможным применение одного или некоторого универсального метода поиска решения для всех вариантов КИНС ППР и МАС.</p> <p>Предложение автора диссертационной работы состоит в том, что в рамках ОМИА отдельные компоненты модели должны рассматриваться как библиотеки моделей, позволяющие создавать различные конкретные модели ИА.</p> <p>С точки зрения реализации действий известные модели следует разбить на три класса:</p> <p>1) с предопределенным конечным множеством элементарных действий [209], [211], [272], [273], [303]-[305];</p>
<p>С. 87.</p>	<p>С. 164.</p>
<p>2) з множиною планів [118–119];</p> <p>3) з довільними повідомленнями і діями в логічній мові [120–121].</p> <p>Узагальнюючи ці моделі пошуку рішення, в узагальненій моделі ІА пропонується наступний варіант пошуку рішення.</p> <p>Вважаємо, що ІА має визначену множину статичних цілей $GS \{gs^i i = 1, \dots, n\}$. Априорі відомі шляхи досягнення цілей, тобто побудовані інформаційні об'єкти $(IO^i i = 1, \dots, n)$, функціонування яких повинне вести до gs^i. Тут кожен ІО покриває деякий план, слідуючи термінології [122–124]. Усередині ж цього плану, тобто в моделі поведінки ІО, можуть бути сформовані довільні повідомлення і довільні послідовності дій, як показано у розд. 2.1.</p>	<p>2) с множеством планов [155], [227], [270], [271];</p> <p>3) с произвольными сообщениями и действиями в логическом языке [269], [277], [283].</p> <p>Обобщая эти модели поиска решения, в ОМИА автор предлагает вариант модели MSRi, определяемый следующим образом.</p> <p>Полагаем, что ИА имеет предопределенное множество статических целей $GS = \{gs^i i = 1, \dots, n\}$. Априорно известны пути достижения целей, т. е. построены информационные объекты $(IO^i i = 1, \dots, n)$, функционирование которых должно вести к gs^i. Здесь каждый ИО покрывает некоторый план, следуя терминологии [155], [227], [270], [271]. Внутри же этого плана, т. е. в модели поведения ИО, могут быть сформированы произвольные сообщения и произвольные последовательности действий, как показано в разд. 2.2.4. и 2.2.5.</p>
<p>С. 87.</p>	<p>С. 164–165.</p>
<p>Тоді модель пошуку рішення задається функцією пошуку рішення $SR : GS \rightarrow VO$ де VO – множина вкладених ІО і-го ІА. Це відображення однозначне, але не взаємно, оскільки можливо, що декілька цілей досягаються одним і тим же ІО. Модель активних дій визначається відображенням $AD : GA \rightarrow VO$, яке вибирає необхідні для запуску у нинішній момент ІО. Процес запуску ІО реалізується на стадії функціонування ІКАС. Зміна активних цілей (зміна множини GA) приводить до зупинки і запуску відповідних ІО.</p>	<p>Тогда модель поиска решения задается функцией поиска решения $SR : GS \rightarrow VO$, где VO - множество вложенных ИО i-го ИА. Это отображение однозначное, но не взаимно, так как возможно, что несколько целей достигаются одним и тем же ИО. Модель активных действий определяется отображением $AD : GA \rightarrow VO$, которое выбирает необходимые для запуска в текущий момент ИО. Процесс запуска ИО реализуется на стадии исполнения КИНС ППР. Смена активных целей (изменение множества GA) будет приводить к остановке и запуску соответствующих ИО.</p>
<p>С. 87–88.</p>	<p>С. 165.</p>
<p>Далі розробляється продукційна формалізація функціонування інтелектуального агента в узагаль-</p>	<p>Далее разрабатывается продукционная формализация функционирования интеллектуального</p>

<p>неній моделі ІА. Введемо наступні позначення: SA(0) – початкова конфігурація структури атрибутів ІА, Lvo – список вкладених ІО, SMA(0) – початкова конфігурація МА-оточення, Ls – список стратегій ІА, Lsg – список статичних цілей, Ltg – список цілей, що отримуються “зверху”, Ldg – список цілей, передаваних “вниз”, Ldag – список автономних динамічних цілей, Lag – список активних цілей.</p>	<p>агента в ОМІА. Введем следующие обозначения: SA(0) – начальная конфигурация структуры атрибутов ИА, lvo - список вложенных ИО, SMA(0) – начальная конфигурация МА-мира, ls - список стратегий ИА, lsg - список статических целей, Ug - список целей, получаемых «сверху», ldg - список целей, передаваемых «вниз», ldag - список автономных динамических целей, lag - список активных целей.</p>
<p>С. 88.</p> <p>Аксиома числення KMVB у момент первинного запуску ІА приймає вигляд</p> $AA = \{SA(0); Lvo; SMA(0); Ls(0); Lsg(0); \emptyset; \emptyset; \emptyset; \emptyset\}$	<p>С. 165.</p> <p>Аксиома исчисления KMVB в момент первоначального запуска ИА принимает вид</p> $AA = \begin{matrix} SA(0) \\ lvo \\ SMA(0) \\ ls(0) \\ lsg(0) \\ \emptyset \\ \emptyset \\ \emptyset \\ \emptyset \end{matrix} .$
<p>С. 88.</p> <p>Для спрощення запису інформаційного простору V і його стану SV введемо</p> $Lia = \{NIA_j, \{A_{IA_j}^\psi \psi = 1, \dots, k\} j = 1, \dots, l\},$ $Lio = \{NIO_j, \{A_{IO_j}^\psi \psi = 1, \dots, k\} j = 1, \dots, l\}.$ <p>V представляємо двома списками – Lia, Lio, а SV – списками атрибутів Laia, Laio. Тоді $(V, SV) \Leftrightarrow Lia, Lio, Laia, Laio$.</p>	<p>С. 165.</p> <p>Для упрощения записи информационного пространства V и его состояния SV введем</p> $lia = \{NIA_j, \{A_{IA_j}^\psi \psi = 1, \dots, k\} j = 1, \dots, l\}$ $lio = \{NIO_j, \{A_{IO_j}^\psi \psi = 1, \dots, k\} j = 1, \dots, l\}.$ <p>V представляем двумя списками - lia, lio, а SV - списками атрибутов - laia, laio. Тогда $(V, SV) \Leftrightarrow lia, lio, laia, laio$.</p>
<p>С. 88.</p> <p>Формування першої ситуації МА-оточення описуємо правилом</p> $(1) AA \Rightarrow (2)Lia, Lio, \emptyset, \emptyset; AA.$ <p>Виконання обсервації визначиться правилом (2)</p> $(2)Lia, Lio, \emptyset, \emptyset; AA \Rightarrow (3)FV(lia, lio, \emptyset, \emptyset); AA.$	<p>С. 165–166.</p> <p>Формирование первой ситуации МА-мира описываем правилом</p> $(1) \frac{AA}{(2)lia, lio, \emptyset, \emptyset}.$ <p>Выполнение обсервации определится правилом (2)</p> $(2) \frac{AA}{(3)FV(lia, lio, \emptyset, \emptyset)}.$
<p>С. 88–89.</p> <p>Далі для скорочення запису позначатимемо складові антецедента кутовими дужками з нижнім індексом як номер, і, якщо вміст даного компоненту не змінюється в даному правилі, то просто повторюватимемо Ri . В правилах числення приписуємо на початок антецедента номер правила в круглих дужках (...), а в консеквенті формуватимемо номер наступного правила.</p> <p>Тоді формування початкового стану МА-оточення опишеться правилом</p> $(3) Lia, Lio, Laia, Laio; AA \Rightarrow (4) \mathfrak{R}_1 \langle SA \rangle_2 \langle lvo \rangle_3 \langle \emptyset; Lia' := FP(\mathfrak{R}_1), Lio' := FP(\mathfrak{R}_1), Laia' := FP(\mathfrak{R}_1), Laio' := FP(\mathfrak{R}_1) \rangle_4 \langle ls \rangle_5 \langle lsg \rangle_6 \langle \emptyset \rangle_7 \langle \emptyset \rangle_8 \langle \emptyset \rangle_9 \langle \emptyset \rangle_{10}$	<p>С. 166.</p> <p>Далее для сокращения записи будем обозначать составляющие антецедента угловыми скобками с нижним индексом в качестве номера, и если содержимое данного компонента не изменяется в данном правиле, то будем просто повторять [...]i. Тогда формирование начального состояния МА-мира опишется правилом (3)</p> $(3) \frac{AA}{(4) \langle \dots \rangle_1 \langle SA \rangle_2 \langle lvo \rangle_3 \langle \emptyset; lia' := FP(\dots)_1, lio' := FP(\dots)_1, laia' := FP(\dots)_1, laio' := FP(\dots)_1 \rangle_4 \langle ls \rangle_5 \langle lsg \rangle_6 \langle \emptyset \rangle_7 \langle \emptyset \rangle_8 \langle \emptyset \rangle_9 \langle \emptyset \rangle_{10}}$
<p>С. 89.</p> <p>Вибір стратегії задається правилом (4)</p>	<p>С. 166.</p> <p>Выбор стратегии задается правилом (4)</p>

<p>(4) $\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4(1s)_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10} \Rightarrow$ $\Rightarrow (5) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4\langle s' := FSS(1s, \mathfrak{R}_4, \mathfrak{R}_{10}); 1s \rangle_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}$ Для прийому цілей, що отримуються “зверху” введемо операцію читання буфера “верхніх” цілей – read(blfg). Відповідне правило приймає вигляд (5) $\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10} \Rightarrow$ $\Rightarrow (6) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\langle Ltg := read(blfg) \rangle_8\mathfrak{R}_9\mathfrak{R}_{10}$</p>	<p>(4) $\frac{(4)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}}{(5)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}}$ Для приёма целей, получаемых «сверху» введем операцию чтения буфера «верхних» целей – read(blfg). Соответствующее правило принимает вид (5) $\frac{(5)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}}{(6)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}}$</p>
<p>C. 89–90.</p>	<p>C. 166–167.</p>
<p>Далі можна сформулювати цілі передавані “вниз”, динамічні цілі з множини GAVT(D) і вибрати статичні цілі (6) $\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10} \Rightarrow$ $\Rightarrow (7) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\langle Ldg := h^{DOWN}(\mathfrak{R}_4, s', U, Lag) \rangle_8$ $\langle Ldag := h^{AVT(D)}(\mathfrak{R}_4, s', U, Lag) \rangle_9\mathfrak{R}_{10}\langle Lavtgs := h^{AVT(D)}(\mathfrak{R}_4, s', U, Lag) \rangle_{11}$</p>	<p>Далее можно сформировать цели передаваемые «вниз», динамические цели из множества GAVT(D) и выбрать статические цели GAVT(S) (6) $\frac{(6)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}}{(7)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}}$ $[ldag := h^{AVT(D)}([\dots]_4, s', U, lag)]_9[\dots]_{10}[lavgts := h^{AVT(D)}([\dots]_4, s', U, lag)]_{11}$</p>
<p>C. 90.</p>	<p>C. 166–167.</p>
<p>Тепер може бути побудована множина активних цілей (7) $\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (8) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8$ $\mathfrak{R}_9\langle Lag := h^A(s', \mathfrak{R}_9, \mathfrak{R}_{10}, \mathfrak{R}_{11}, Z) \rangle_{10}\mathfrak{R}_{11}$</p>	<p>Теперь может быть построено множество активных целей (7) $\frac{(7)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}[\dots]_{11}}{(8)[\dots]_1[\dots]_2[\dots]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}[\dots]_{11}}$ $[\dots]_9[lag := h^A(s', [\dots]_9, [\dots]_{10}, [\dots]_{11}, Z)]_{10}[\dots]_{11}$</p>
<p>C. 90.</p>	<p>C. 167.</p>
<p>Для кожної активної мети gaJ вибираємо підмножину IO для яких априорі припускаємо, що вони можуть вести до даної активної мети. Вважаємо, що цілі, що містяться в Lag не суперечливі, виконання цієї вимоги повинне забезпечити перетворення hA що використовує онтологію активізації цілей Z. Отже, множина VO розбивається на n непересічних підмножин, таких що $VO \subseteq \bigcup_{j=1}^n VOga^j.$</p>	<p>Для каждой активной цели gaJ выбираем подмножество IO для которых априори предполагаем, что они могут вести к данной активной цели. Считаем, что цели, содержащиеся в lag не противоречивы, выполнение этого требования должно обеспечить преобразование hA, использующее онтологию активизации целей Z. Следовательно, множество VO разбивается на n непересекающихся подмножеств, таких что $VO \subseteq \bigcup_{j=1}^n VOga^j.$</p>
<p>C. 90–91.</p>	<p>C. 167.</p>
<p>Такі підмножини VOga^j називатимемо модельною множиною вкладених IO і позначати ModelSet^j. IO, що входять в j-ту модельну множину позначатимемо IO_{m^j}, де нижній індекс t показує, що даний IO моделюється в процесі пошуку рішення. Для здійснення такого розбиття потрібна функція FMVO = h VO (Lvo, Lag, X), де X – онтологія, що описує логіку такого розбиття і необхідна тому, що потенційна множина динамічних цілей може бути нескінченною. Формулюємо правило (8) $\mathfrak{R}_1\mathfrak{R}_2\langle lvo \rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\langle Lag \neq \emptyset \rangle_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1\mathfrak{R}_2\langle h^A(Lvo, Lag, X) \rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\langle Lag \neq \emptyset \rangle_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (9) \mathfrak{R}_1\mathfrak{R}_2\langle ModelSet, Lvo \rangle_3\mathfrak{R}_4\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p>	<p>Такие подмножества VOga^j будем называть модельными множествами вложенных IO и обозначать ModelSet^j. Входящие в j-е модельное множество IO будем обозначать IO_{m^j}, где нижний индекс t показывает, что данный IO моделируется в процессе поиска решения. Для осуществления такого разбиения требуется функция FMVO = h vO(lvo, lag, X), где X - онтология, описывающая логику такого разбиения, необходимая потому, что потенциальное множество динамических целей может быть бесконечно. Формулируем правило (8) $\frac{(8)[\dots]_1[\dots]_2[lvo]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[lag \neq \emptyset]_{10}[\dots]_{11}}{[\dots]_1[\dots]_2[h^A(lvo, lag, X)]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[lag \neq \emptyset]_{10}[\dots]_{11}}$ $\frac{(9)[\dots]_1[\dots]_2[ModelSet, lvo]_3[\dots]_4[\dots]_5[\dots]_6[\dots]_7[\dots]_8[\dots]_9[\dots]_{10}[\dots]_{11}}$</p>
<p>C. 91.</p>	<p>C. 167–168.</p>
<p>Інформаційні об’єкти відповідно до своєї моделі поведінки формують ПІО (пасивні інформаційні об’єкти) і передають їх як повідомлення; а також виконують перетворення hA над структурою своїх атрибутів. Тому в процесі пошуку рішення ми моделюємо поведінку всіх IO, що належать даній модельній множині для пошуку шляху до мети gaJ на глибину в gr кроків. Вважаємо, що параметр gr визначається ви-</p>	<p>Информационные объекты в соответствии со своей моделью поведения формируют ПИО(пасивные информационные объекты) и передают их как сообщения, а также выполняют преобразования hA над структурой своих атрибутов. Поэтому в процессе поиска решения мы моделируем поведение всех IO, принадлежащих данному модельному множеству для поиска пути к цели gaJ на глубину в gr шагов. Полагаем, что параметр gr определяет-</p>

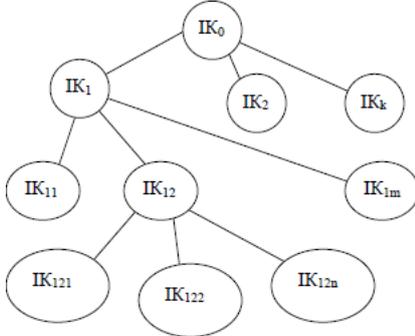
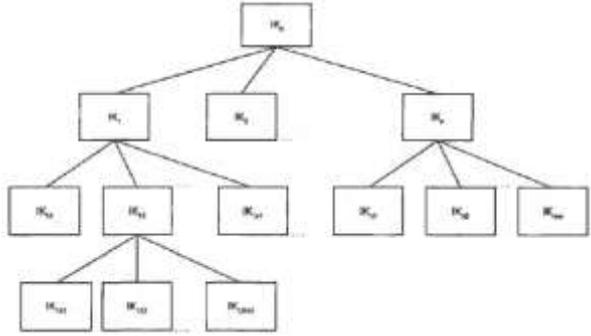
<p>браною стратегією si і вводимо два лічильники: лічильник модельних кроків $Stgr$ і лічильник актуальних кроків $CaGr$. Виконуємо модельні кроки для модельних множин на глибину gr. Після кожного кроку обчислюємо передбачувану оцінку стану МА-оточення для всіх модельованих ІО і робимо наступний модельний крок. Повторюємо модельні кроки до досягнення $Stgr = gr$.</p> <p>Далі обчислюємо передбачувану оцінку перетворень цілей для всіх ІО і відхилення від цілей (міру близькості до мети). Визначаємо найменше відхилення</p> $\Delta_{min} = \min(\Delta(IO_m^{j,i}) IO_m^{j,i} \in ModelSet^j).$	<p>ся выбранной стратегией si и вводим два счетчика: счетчик модельных шагов $Stgr$ и счетчик актуальных шагов $CaGr$. Выполняем модельные шаги для модельных множеств на глубину gr. После каждого шага вычисляем предполагаемую оценку состояния МА-мира для всех моделируемых ИО и делаем следующий модельный шаг. Повторяем модельные шаги до достижения $Stgr = gr$.</p> <p>Далее вычисляем предполагаемую оценку преобразований целей для всех ИО и отклонение от целей (меру близости к цели). Определяем наименьшее отклонение</p> $\Delta_{min} = \min(\Delta(IO_m^{j,i}) IO_m^{j,i} \in ModelSet^j).$
<p>C. 91–92.</p>	<p>C. 168.</p>
<p>Ці дії формалізуємо правилами (9)</p> <p>(9) $\mathfrak{R}_1 \mathfrak{R}_2 \langle LModelSet, Lvo \rangle_3 \mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1 \mathfrak{R}_2 \langle LModelSet, Lvo \rangle_3 \mathfrak{R}_4 \langle s', Ls, Cmgp \Rightarrow h^{SP}(s'); CaGr \Rightarrow h^{SP}(s') \rangle_5$ $\mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1 \mathfrak{R}_2 \langle LModelSet(K_{3M}(IO_m^{j,i}) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), Lvo \rangle_3$ $\mathfrak{R}_4 \langle s', Ls, Cmgp = Cmgp - 1, CaGr \rangle_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1 \mathfrak{R}_2$ $\langle LModelSet(K_{3M}(IO_m^{j,i}), FP(R, K_{3M}(IO_m^{j,i}) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), Lvo) \rangle_3$ $\mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1 \mathfrak{R}_2$ $\langle LModelSet(K_{3M}(K_{3M}(IO_m^{j,i}), R_i) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), Lvo) \rangle_3$ $\mathfrak{R}_4 \langle s', Ls, Cmgp = Cmgp - 1, CaGr \rangle_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow \dots \Rightarrow \dots \Rightarrow$ $\Rightarrow \mathfrak{R}_1 \mathfrak{R}_2 \langle LModelSet(K_{3M}(K_{3M}(IO_m^{j,i}), R_i) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), Lvo) \rangle_3$ $\mathfrak{R}_4 \langle s', Ls, Cmgp = 0, CaGr \rangle_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow \mathfrak{R}_1 \mathfrak{R}_2$ $\langle LModelSet(h^{SP}(K_{3M}(K_{3M}(IO_m^{j,i}), R_i) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), Lvo) \rangle_3$ $\mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1 \mathfrak{R}_2 \langle LModelSet(\Delta(gn^j, gn^j) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), Lvo) \rangle_3$ $\mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (10) \mathfrak{R}_1 \mathfrak{R}_2 \langle LModelSet(\Delta_{min} = \min(\Delta(gn^j, gn^j)) j=1, \dots, n), Lvo) \rangle_3$ $\mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11}$</p>	<p>Эти действия формализуем правилами (9)</p> <p>(9) $\frac{[9] \dots [1] \dots [3] [ModelSet, lvo]_3 [1] \dots [4] \dots [5] [s', Ls, Cmgp \Rightarrow h^{SP}(s'); CaGr \Rightarrow h^{SP}(s')]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{[1] \dots [2] [ModelSet, lvo]_3 [1] \dots [4] [s', Ls, Cmgp \Rightarrow h^{SP}(s'); CaGr \Rightarrow h^{SP}(s')]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]} \Rightarrow$ $\frac{[1] \dots [2] [ModelSet(K_{3M}(IO_m^{j,i}) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp \Rightarrow Cmgp - 1, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{[1] \dots [2]}$ $\frac{[ModelSet(K_{3M}(IO_m^{j,i}), FP(R, K_{3M}(IO_m^{j,i}) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = Cmgp - 1, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{[1] \dots [2] [ModelSet(K_{3M}(IO_m^{j,i}), FP(R, K_{3M}(IO_m^{j,i}) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = Cmgp - 1, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]} \Rightarrow$ \dots $\frac{[1] \dots [2] [ModelSet(K_{3M}(K_{3M}(IO_m^{j,i}), R_i) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{[1] \dots [2] [ModelSet(K_{3M}(K_{3M}(IO_m^{j,i}), R_i) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]} \Rightarrow$ \dots $\frac{[1] \dots [2] [ModelSet(h^{SP}(K_{3M}(K_{3M}(IO_m^{j,i}), R_i) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{[1] \dots [2] [ModelSet(h^{SP}(K_{3M}(K_{3M}(IO_m^{j,i}), R_i) IO_m^{j,i} \in ModelSet^j, j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]} \Rightarrow$ $\frac{(10) [1] \dots [2] [ModelSet(\Delta_{min} = \min(\Delta(gn^j, gn^j)) j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{(10) [1] \dots [2] [ModelSet(\Delta_{min} = \min(\Delta(gn^j, gn^j)) j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]} \Rightarrow$</p>
<p>C. 92–93.</p>	<p>C. 169.</p>
<p>Після цього кроку слід сформуванати множину актуальних ІО, які діятимуть і посилатимуть повідомлення, таким чином, переходячи до фази активних дій і реалізації моделі МА:</p> <p>(10) $\mathfrak{R}_1 \mathfrak{R}_2 \langle (\Delta_{min}^j j=1, \dots, n), lvo \rangle_3 \mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1 \mathfrak{R}_2 \langle (IO_m^{1,1}, \dots, IO_m^{1,mm}, \dots, IO_m^{1,k_1}) (IO_m^{2,1}, \dots, IO_m^{2,mm}, \dots, IO_m^{2,k_2}) \dots$ $(IO_m^{n,1}, \dots, IO_m^{n,mm}, \dots, IO_m^{n,k_n}) : lvo \rangle_3 \mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (11) \mathfrak{R}_1 \mathfrak{R}_2 \langle lvo = h^R(lvo) \rangle_3 \mathfrak{R}_4 \mathfrak{R}_5 \mathfrak{R}_6 \mathfrak{R}_7 \mathfrak{R}_8 \mathfrak{R}_9 \mathfrak{R}_{10} \mathfrak{R}_{11}$</p>	<p>После этого шага следует сформировать множество актуальных ИО, которые будут действовать и посылать сообщения, таким образом, переходя к фазе активных действий и реализации модели МА:</p> <p>(10) $\frac{(10) [1] \dots [2] [(\Delta_{min}^j j=1, \dots, n), lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{[1] \dots [2] [(IO_m^{1,1}, \dots, IO_m^{1,mm}, \dots, IO_m^{1,k_1}) (IO_m^{2,1}, \dots, IO_m^{2,mm}, \dots, IO_m^{2,k_2}) \dots (IO_m^{n,1}, \dots, IO_m^{n,mm}, \dots, IO_m^{n,k_n}) : lvo]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]} \Rightarrow$ $\frac{(11) [1] \dots [2] [lvo = h^R(lvo)]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]}{(11) [1] \dots [2] [lvo = h^R(lvo)]_3 [1] \dots [4] [s', Ls, Cmgp = 0, CaGr]_5 [1] \dots [6] \dots [7] \dots [8] \dots [9] \dots [10] \dots [11]} \Rightarrow$</p>
<p>C. 93.</p>	<p>C. 169.</p>
<p>Зміст цього правила полягає в тому, що ІО, відповідні $\Delta jmin$ позначаються як актуальні, а решта $IO_{m,j,i}$ заміщають своїх батьків в списку Lvo за допомогою перетворення зворотної підстановки hR. Вміст компонента $R3$ приймає вигляд $IO1, \dots, IOia, \dots, IOja, \dots, IOl$, де l – потужність множини вкладених інформаційних об'єктів $\{VO\}$.</p> <p>Далі включаємо в цю множину і виділений інформаційний об'єкт $IO0$, що представляє дії IA над своїми атрибутами, який активізується за умовчанням.</p> <p>Описуємо фазу активних дій наступним правилом:</p>	<p>Смысл этого правила состоит в том, что ИО, соответствующие $\Delta jmin$ помечаются как актуальные, а остальные $IO_{m,j,i}$ замещают своих родителей в списке lvo посредством преобразования обратной подстановки hR. Содержимое компонента $[...]\{3$ принимает вид</p> $IO_1, \dots, IO_i^a, \dots, IO_j^a, \dots, IO_l,$ <p>где l - мощность множества вложенных информационных объектов $\{VO\}$.</p> <p>Далее включаем в это множество и выделенный информационный объект $IO0$, представляющий действия IA над своими атрибутами, который активизируется по умолчанию.</p>

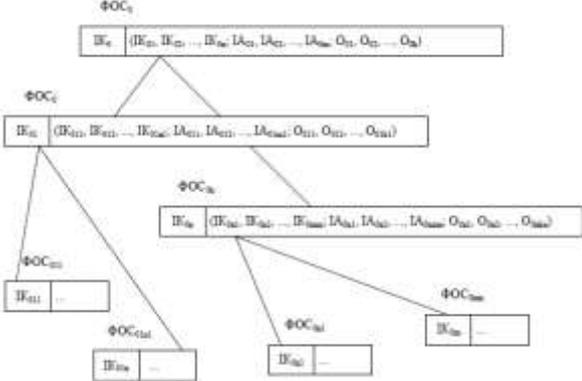
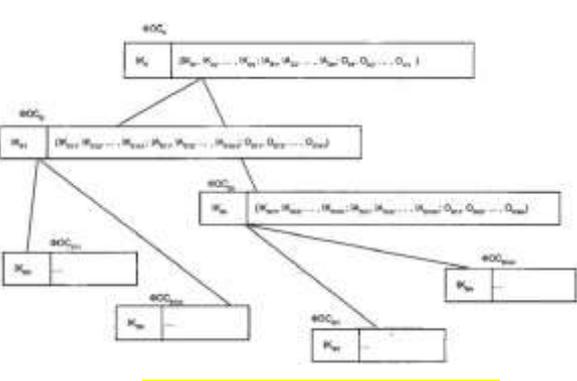
<p>(11) $\mathfrak{R}_1\mathfrak{R}_2\langle IO_0^a, \dots, IO_j^a, \dots \rangle_3 \mathfrak{R}_4\langle s'; Ls; Cagg \rangle_5 \mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1\mathfrak{R}_2\langle K_{МП}(IO_0^a), \dots, K_{МП}(IO_j^a), \dots \rangle_3 \mathfrak{R}_4\langle s'; Ls; Cagg \equiv Cagg - 1 \rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow \dots \Rightarrow \dots \Rightarrow \dots$ $\Rightarrow (12) \mathfrak{R}_1\mathfrak{R}_2\langle K_{МП}(K_{МП} \dots (IO_0^a)), \dots, K_{МП}(K_{МП} \dots (IO_j^a)) \dots \rangle_3 \mathfrak{R}_4$ $\langle s'; Ls; Cagg = 0 \rangle_5 \mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p>	<p>Описуємо фазу активних дій наступним правилом:</p> <p>(11) $\frac{(11) \frac{[\dots]_1 [\dots]_2 [IO_0^a, \dots, IO_j^a, \dots]_3 [\dots]_4 [s'; Ls; Cagg]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{[\dots]_1 [\dots]_2 [K_{МП}(IO_0^a), \dots, K_{МП}(IO_j^a), \dots]_3 [\dots]_4 [s'; Ls; Cagg \equiv Cagg - 1]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}}{[\dots]_1 [\dots]_2 [K_{МП}(K_{МП} \dots (IO_0^a)), \dots, K_{МП}(K_{МП} \dots (IO_j^a)) \dots]_3 [\dots]_4 [s'; Ls; Cagg = 0]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}}$</p>
<p>С. 93–94.</p>	<p>С. 169–170.</p>
<p>Після того, як активні дії закінчилися, можна виконати перевірку і зняти мітки активності з ІО:</p> <p>(12) $\langle Lia, Lio, Laia, Laio \rangle_1 \mathfrak{R}_2 \langle IO_0^a, \dots, IO_j^a, \dots \rangle_3 \mathfrak{R}_4 \langle s'; Ls; Cagg = 0 \rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (13) \langle Lia', Lio', Laia', Laio' \rangle_1 \mathfrak{R}_2 \langle Pa; Rt; Fu \rangle_4 \langle s'; Ls \rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (14) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3$ $\langle Rt; Pa; Lia', Lio', Laia', Laio'; Fu \equiv FP(Lia', Lio', Laia', Laio') \rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p> <p>Формуємо новий поточний стан МА-оточення за допомогою правил (13) і (14)</p> <p>(13) $\langle Lia', Lio', Laia', Laio' \rangle_1 \mathfrak{R}_2 \langle Pa; Rt; Fu \rangle_4 \langle s'; Ls \rangle_5$ $\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow (14) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3$ $\langle Rt; Pa; Lia', Lio', Laia', Laio'; Fu \equiv FP(Lia', Lio', Laia', Laio') \rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p> <p>(14) $\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa; Rt; Fu \rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle ga^j \equiv h^{ga^j} (Pa, Rt, Fu) j = 1, \dots, n; Pa; Rt; Fu \rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (15) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta(ga^j, g\hat{a}^j) j = 1, \dots, n; Pa; Rt; Fu \rangle_4$ $\mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p>	<p>Після того, як активні дії закончилися, можна виконати обсервацію і зняти мітки активності з ІО:</p> <p>(12) $\frac{(12) \frac{[Lia, Lio, Laia, Laio]_1 [\dots]_2 [IO_0^a, \dots, IO_j^a, \dots]_3 [\dots]_4 [s'; Ls; Cagg = 0]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{(13) \frac{[Lia', Lio', Laia', Laio']_1 [\dots]_2 [Pa; Rt; Fu]_4 [\dots]_3 [IO_0^a, \dots, IO_j^a, \dots]_3 [\dots]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{[\dots]_1 [\dots]_2 [\dots]_3 [\dots]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}}$</p> <p>Формуємо нове текущее состояние МА-мира с помощью правил (13) и (14)</p> <p>(13) $\frac{[Lia', Lio', Laia', Laio']_1 [\dots]_2 [Pa; Rt; Fu]_4 [\dots]_3 [s'; Ls]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{(14) [\dots]_1 [\dots]_2 [\dots]_3 [Rt; Pa; Lia', Lio', Laia', Laio'; Fu \equiv FP(Lia', Lio', Laia', Laio')]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}$</p> <p>(14) $\frac{(14) [\dots]_1 [\dots]_2 [\dots]_3 [Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{[\dots]_1 [\dots]_2 [\dots]_3 [ga^j \equiv h^{ga^j} (Pa, Rt, Fu) j = 1, \dots, n; Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}$</p> <p>(15) $\frac{(15) [\dots]_1 [\dots]_2 [\dots]_3 [\Delta(ga^j, g\hat{a}^j) j = 1, \dots, n; Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}$</p>
<p>С. 94–95.</p>	<p>С. 170.</p>
<p>Обчислюємо оцінки стану цілей від досягнутого стану МА-оточення і визначаємо відхилення від мети. Оскільки цілей може бути декілька (у разі $j > 1$), необхідно сформулювати комплексну оцінку положення ІА в просторі цілей</p> $\Delta_k = \sum_{j=1}^n a_j \Delta(ga^j, g\hat{a}^j),$ <p>що враховує значущість цілей, a_i – вагові коефіцієнти важливості цілей, n – потужність множини активних цілей. Тому вводимо правило (15)</p> <p>(15) $\mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta(ga^j, g\hat{a}^j) j = 1, \dots, n; Pa; Rt; Fu \rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow \forall \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta_k; Pa; Rt; Fu \rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p>	<p>Вычисляем оценки состояния целей от достигнутого состояния МА-мира и определяем отклонения от цели. Поскольку целей может быть несколько (в случае $j > 1$), необходимо сформировать комплексную оценку положения ИА в пространстве целей</p> $\Delta_k = \sum_{j=1}^n a_j \Delta(ga^j, g\hat{a}^j),$ <p>учитывающую значимость целей, a_i – весовые коэффициенты важности целей, n – мощность множества активных целей. Поэтому вводим правило (15)</p> <p>(15) $\frac{(15) [\dots]_1 [\dots]_2 [\dots]_3 [\Delta(ga^j, g\hat{a}^j) j = 1, \dots, n; Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{\forall [\dots]_1 [\dots]_2 [\dots]_3 [\Delta_k; Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}$</p>
<p>С. 95.</p>	<p>С. 171.</p>
<p>Проводимо порівняння з допустимим комплексним відхиленням Δd за допомогою наступних правил:</p> <p>(16) $\forall \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta_k > \Delta d; Pa; Rt; Fu \rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (4) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa; Rt; Fu \rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p> <p>(17) $\forall \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle \Delta_k \leq \Delta d; Pa; Rt; Fu \rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11} \Rightarrow$ $\Rightarrow (5) \mathfrak{R}_1\mathfrak{R}_2\mathfrak{R}_3 \langle Pa; Rt; Fu \rangle_4 \mathfrak{R}_5\mathfrak{R}_6\mathfrak{R}_7\mathfrak{R}_8\mathfrak{R}_9\mathfrak{R}_{10}\mathfrak{R}_{11}$</p>	<p>Проводим сравнение с допустимым комплексным отклонением Δd с помощью следующих правил:</p> <p>(16) $\frac{\forall [\dots]_1 [\dots]_2 [\dots]_3 [\Delta_k > \Delta d; Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{(4) [\dots]_1 [\dots]_2 [\dots]_3 [Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}$</p> <p>(17) $\frac{\forall [\dots]_1 [\dots]_2 [\dots]_3 [\Delta_k \leq \Delta d; Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}{(5) [\dots]_1 [\dots]_2 [\dots]_3 [Pa; Rt; Fu]_4 [\dots]_5 [\dots]_6 [\dots]_7 [\dots]_8 [\dots]_9 [\dots]_{10} [\dots]_{11}}$</p>
<p>С. 95.</p>	<p>С. 171.</p>
<p>При виконанні умови $\Delta_k > \Delta d$ переходимо до</p>	<p>При выполнении условия $\Delta_k > \Delta d$ переходим к</p>

<p>правила (4), тобто переоцінюємо стратегію формування цілей; якщо ж $\Delta k \leq \Delta d$ то функціонування ІА продовжується у тій же самій стратегії.</p> <p>Побудована в даному розділі модель пошуку рішення в узагальненій моделі ІА дозволяє описати такі відомі класи моделей реалізації поведінки як моделі із зумовленою кінцевою множиною елементарних дій; моделі з множиною планів; моделі з довільними повідомленнями і діями. На базі даної моделі можуть створюватися нові моделі реалізації поведінки ІА, що поєднують механізми різних класів.</p>	<p>правилу (4), т. е. переоцениваем стратегию формирования целей; если же $\Delta k \leq \Delta d$, то продолжается функционирование ИА в той же стратегии. В правилах нашего исчисления приписываем к началу antecedenta номер правила в круглых скобках (...), а в консеквенте будем формировать номер следующего правила.</p> <p>Построенная в данном разделе модель поиска решения в ОМИА позволяет описать такие известные классы моделей реализации поведения как модели с predetermined конечным множеством элементарных действий; модели с множеством планов; модели с произвольными сообщениями и действиями. На базе данной модели могут создаваться новые модели реализации поведения ИЛ, сочетающие механизмы различных классов.</p>
<p>С. 95–96.</p>	<p>С. 171.</p>
<p>Розроблена узагальнена модель ІА може слугувати теоретичним фундаментом для побудови широкого класу агентних моделей ІКСА, що відрізняються моделями інформаційного простору, моделями цілевизначення, моделями пошуку рішення і моделями активних дій.</p> <p>На основі узагальненої моделі ІА на базі однієї і тієї ж ІКСА можна досліджувати і створювати різноманітні мультиагентні системи, які включають ІА різних класів і видів і здатні ефективно “обслуговувати” різних посадових осіб авіапідприємства.</p>	<p>Разработанная автором обобщенная модель ИА может служить теоретическим фундаментом для построения новых видов агентных моделей, отличающихся моделями информационного пространства, моделями целеполагания, моделями поиска решения и моделями активных действий.</p> <p>На основе обобщенной модели ИА можно исследовать и создавать гетерогенные МАС и ИНС, включающие И А различных классов и видов, и обладающие потенциально неограниченным разнообразием.</p>
<p>С. 96–97.</p>	<p>С. 172.</p>
<p>Висновки до розділу 2</p> <p>Завдання інтелектуалізації інформаційно-керуючої системи аеропорту найбільш доцільно вирішити шляхом запровадження у програмне забезпечення системи інтелектуальних компонентів – інформаційних об’єктів та інтелектуальних агентів.</p> <p>Аналіз відомих теоретичних моделей, що склалися в об’єктно-орієнтованому проектуванні і програмуванні, показав, що дані моделі не дозволяють формалізувати основні властивості інформаційних об’єктів і не задовольняють всім вимогам, що пред’являються до об’єктів, які функціонують у неоднорідному інформаційному середовищі системи управління аеропортом.</p> <p>Тому, на базі моделі скінченого автомата з використанням логіки першого порядку в роботі розроблено модель інформаційного об’єкта, яка дозволяє описувати інформаційні об’єкти в рамках формальної об’єктної системи, що володіє властивістю розширення і замкнутої щодо множини інформаційних об’єктів.</p> <p>Слюняєв пише «Аналіз ... показав», «Тому» і «в роботі розроблено», ніби він робив якийсь аналіз, щось пропонує, щось самостійно розробляв, а насправді переписує чужий текст. Плагіат.</p>	<p>Выводы по главе 2</p> <p><...></p> <p>Анализ известных теоретических моделей, сложившихся в объектно-ориентированном проектировании и программировании, выполненный в разделе 2.2.1, показал, что данные модели не позволяют формализовать основные свойства информационных объектов и не удовлетворяют всем предъявляемым требованиям.</p> <p>Поэтому в разделе 2.2 излагается разработанная автором диссертационной работы модель ИО, основанная на логике первого порядка, позволяющая описывать информационные объекты в рамках формальной объектной системы (ФОС), обладающей свойством расширяемости и замкнутой относительно множества ИО.</p>
<p>С. 97.</p>	<p>С. 172.</p>
<p>(Висновки до розділу 2)</p> <p>Для опису функціонування інформаційного об’єкта найбільш доцільним є продукційна модель, заснована на апараті канонічних числень Е. Поста, що дозволяє створювати різні моделі функціонування інформаційного об’єкта в конкретній системі і забезпечує зв’язок між рівнем програмних об’єктів і</p>	<p>(Выводы по главе 2)</p> <p>Существенным отличием авторской модели ИО является продукционная модель поведения, основанная на аппарате канонических исчислений Э.Л. Поста, позволяющая создавать различные модели поведения ИО в конкретной системе, и обеспечивающая взаимосвязь между уровнем программных</p>

<p>рівнем інтелектуальних агентів в інформаційно-керуючій системі аеропорту.</p> <p>Така інтелектуальна взаємодія забезпечується тим, що в моделі інформаційного об'єкта закладена можливість передачі і прийому повідомлень в довільних контекстно-вільних мовах, передбачені довільні функціональні перетворення над списками атрибутів інформаційного об'єкта і використовується породження і аналіз формул в мові логіки першого порядку.</p>	<p>объектов и уровнем интеллектуальных агентов в КИНС ППР.</p> <p>Такая коммуникация уровней информационной и интеллектуальной интеграции обеспечивается тем, что в модели ИО заложена возможность передачи и приема сообщений в произвольных контекстно-свободных языках, предусмотрены произвольные функциональные преобразования над списками атрибутов ИО, и используется порождение и анализ формул в языке логики первого порядка.</p>
<p>С. 97–98.</p>	<p>С. 173.</p>
<p>(Висновки до розділу 2)</p> <p>Порівняльний аналіз основних типів агентних моделей за умовами їх застосовності до завдань побудови інтелектуальної інформаційно-керуючої системи аеропорту показав, що до теперішнього часу не розроблена достатньо універсальна модель інтелектуального агента, яка враховувала б ієрархію інтелектуальних агентів в мультиагентній системі, структурну ієрархію цілей і, в той же час, була б направлена на технічну реалізацію в рамках існуючих технічних засобів інформаційно-керуючої системи аеропорту.</p> <p>Тому в роботі запропонована узагальнена модель інтелектуального агента в структурі інформаційно-керуючої системи аеропорту, що відрізняється способом формування інформаційного простору інтелектуальних агентів, наявністю моделі механізму вибору поведінки і змістом моделі цільовизначення інтелектуального агента, що дозволяє враховувати динаміку розвитку мультиагентного оточення, складну ієрархію цілей в інформаційно-керуючій системі аеропорту і формувати різні стратегії функціонування інтелектуальних агентів.</p> <p>Слюняєв пише «Порівняльний аналіз ... показав» і «Тому в роботі запропонована», ніби він робив якийсь аналіз, щось пропонує, щось самостійно розробляв, а насправді переписує чужий текст. Плагіат.</p>	<p>(Выводы по главе 2)</p> <p>Выполненный в разделе 2.3.1. сравнительный анализ основных типов агентных моделей по условиям их применимости к задачам построения КИНС ППР показал, что к настоящему времени не разработана достаточно универсальная модель интеллектуального агента, которая учитывала бы иерархию ИА в мультиагентной системе, конструктивно определяла бы цели агента и их иерархию, и в то же время была бы направлена на техническую реализацию.</p> <p>Поэтому в разделе 2.3.2 предложена обобщенная модель интеллектуального агента (ОМИА) в структуре КИНС ППР, базирующаяся на синтезе основных концепций делиберативного и реактивного подходов, отличающаяся способом формирования информационного пространства ИА, наличием модели механизма выбора поведения и содержанием модели целеполагания ИА, позволяющая учитывать динамику развития мультиагентного мира, сложную иерархию целей в КИНС ППР и формировать различные стратегии поведения ИА.</p>
<p>С. 98–99.</p>	<p>С. 244.</p>
<p>3.1. Структурно-логічна модель інтелектуальної інформаційно-керуючої системи аеропорту</p> <p>Основою для аналізу і розробки моделей баз знань в ІІКСА є логічна структура, визначена в розділі 2.3. Необхідно вирішити питання про те, як формально визначити логічну структуру ІІКСА, які математичні засоби доцільно для цього використовувати. Логічна структура ІІКСА повинна розглядатися як структура ієрархічна, з точним визначенням рівнів і підлеглості інтелектуальних компонентів системи.</p> <p>Це положення обумовлюється тим, що структури організаційного управління мають складну ієрархію (оператор – відділ (служба) – управління і т. д.), що визначає адміністративні, виробничо-технічні і економічні зв'язки в проектуваній системі [125]. При цьому, як правило, спостерігається сувора підлеглість адміністративно-структурних одиниць вищим рівням управління.</p>	<p>4.1. Структурно-логіческая модель корпоративної інтелектуальної системи підтримки прийняття рішень</p> <p>Основой для анализа и разработки моделей баз знаний в КИНС ППР является логическая структура, содержательно определенная в разделе 2.3. Необходимо решить вопрос о том, как формально определить логическую структуру КИНС ППР, какие математические средства целесообразно для этого использовать. По мнению автора; логическая структура КИНС ППР должна рассматриваться как структура иерархическая, с точным определением уровней и подчиненности интеллектуальных компонентов системы.</p> <p>Это положение обусловливается тем, что структуры производственного и организационного управления имеют сложную иерархию (участок - цех - завод - отрасль, отдел - департамент - управление и т. д.), определяющую административные, производственно-технические и экономические связи в проектируемой системе [377], [378]. При этом, как правило, наблюдается строгая подчинен-</p>

	<p>Слюняєв переписав чужий текст разом із готовим покликанням, а також видалив інші покликання. Плагіат.</p>	<p>ность административно-структурных единиц вышестоящим уровням управления [А 10], [А14], [А17], [А33].</p>
С. 99.		С. 244.
	<p>Спираючись на основні положення системного аналізу [126], введемо поняття дерева інтелектуальних компонентів (ДІК), під яким розумітимемо зв'язний неорієнтований граф без циклів, вершинами якого є інтелектуальні компоненти ІКСА (що позначаються далі як ІКі ,..., ІКj ,..., ІКіjф,...), а дуги сполучають інтелектуальний компонент з іншими ІК вищого або нижчого рівнів, так що ІК нижчого рівня структурно входить в один з ІК вищого рівня.</p> <p>У дереві ІК можуть бути виділені три типи вершин: Початкова вершина ІК0, яка не має вхідних дуг, відповідає центральному ядру ІКСА. Проміжні вершини – це вершини, що мають строго одну вхідну дугу і n вихідних дуг. Термінальні вершини – це вершини, які мають строго одну вхідну дугу і не мають вихідних дуг, що відповідає ІК найнижчого рівня ієрархії (рівень аеропортових датчиків та сканерів).</p>	<p>Опираясь на основные положения системного анализа [374], введем понятие дерева интеллектуальных компонентов (ДНЮ, под которым будем понимать связный неориентированный граф без циклов, вершинами которого являются интеллектуальные компоненты КИНС ППР (обозначаемые далее как ІКі, ..., ІКj, ..., ІКіjф, ...), а дуги соединяют интеллектуальный компонент с другими ІК вышестоящего или нижестоящего уровней, так что ІК нижестоящего уровня структурно входит в один из ІК вышестоящего уровня.</p> <p>В дереве ІК могут быть выделены три типа вершин. Начальная вершина ІК0, не имеющая входящих дуг, соответствует центральному ядру КИНС ППР. Промежуточные вершины - это вершины, имеющие строго одну входящую дугу и p выходящих дуг. Терминальные вершины - это вершины, имеющие строго одну входящую дугу и не имеющие выходящих дуг, что соответствует ІК самого низкого уровня иерархии.</p>
С. 99–100.		С. 245.
	<p>Множину вершин ДІК – {V} можна розбити на три підмножини: Множина початкової вершини V0, що складається з одного елемента ІК0; множина проміжних вершин {Vp}, множина термінальних вершин {Vt}, так що $\{V\} = \{V0\} \cup \{Vp\} \cup \{Vt\}$.</p> <p>Позначимо зв'язність вершин по входу в ДІК як p_+, а зв'язність вершин по виходу як p_-. Для всіх вершин, окрім ІК0, зв'язність вершин по входу дорівнює одиниці:</p> $\forall V_i ((V_i \in \{V\}) \& (i \neq 0)) \rightarrow (p_+(V_i) = 1).$ <p>Для множин проміжних і термінальних вершин справедливі наступні формули в численні предикатів:</p> $\forall V_i ((p_+(V_i) = 1) \& (p_-(V_i) \neq 0)) \rightarrow (V_i \in \{V_p\});$ $\forall V_i ((p_+(V_i) = 1) \& (p_-(V_i) = 0)) \rightarrow (V_i \in \{V_t\}).$	<p>Множество вершин ДИК - {V} - можно разбить на три подмножества: множество начальной вершины V0, состоящее из одного элемента ІК0; множество промежуточных вершин {Vp}> множество терминальных вершин {Vp}, так что $\{V\} = \{V0\} \cup \{Vp\} \cup \{Vi\}$.</p> <p>Обозначим связность вершин по входу в ДИК как p_+, а связность вершин по выходу как p_-. Для всех вершин, кроме ІК0, связность вершин по входу равна единице:</p> $\forall V_i ((V_i \in \{V\}) \& (i \neq 0)) \rightarrow (p_+(V_i) = 1).$ <p>Для множеств промежуточных и терминальных вершин справедливы следующие формулы в исчислении предикатов:</p> $\forall V_i ((p_+(V_i) = 1) \& (p_-(V_i) \neq 0)) \rightarrow (V_i \in \{V_p\});$ $\forall V_i ((p_+(V_i) = 1) \& (p_-(V_i) = 0)) \rightarrow (V_i \in \{V_t\}).$
С. 100.		С. 245.
	<p>Кількість вершин в ДІК визначається рівністю $NV = Np + Nt + 1$, де Np – кількість проміжних вершин, Nt – кількість ТЕРМІНАЛЬНИХ вершин. Ці співвідношення можуть бути використані на етапі декомпозиції ІКСА для перевірки коректності побудови ДІК.</p> <p>Використовуючи введені позначення, графічне уявлення ДІК прийме вигляд, поданий на рис. 3.1. У лінійному вигляді ДІК можна відобразити списком за рівнями ієрархії:</p> $IK_0 (IK_1, IK_2, \dots, IK_n); IK_1 (IK_{11}, IK_{12}, \dots, IK_{1n_1}); \dots$ $IK_n (IK_{n1}, IK_{n2}, \dots, IK_{nm_n}); IK_{11} (IK_{111}, IK_{112}, \dots, IK_{11n_1}); \dots$	<p>Количество вершин в ДИК определяется равенством $NV = Np + Nt + 1$, где Np - количество промежуточных вершин, Nt - количество терминальных вершин. Эти соотношения могут быть использованы на этапе декомпозиции КИНС ППР для проверки корректности построения ДИК.</p> <p>Используя введенные обозначения, графическое представление ДИК примет вид, представленный на рис. 4.1. В линейном виде ДИК можно отобразить списком по уровням иерархии:</p> $IK_0 (IK_1, IK_2, \dots, IK_n);$ $IK_1 (IK_{11}, IK_{12}, \dots, IK_{1n_1});$ \dots $IK_n (IK_{n1}, IK_{n2}, \dots, IK_{nm_n});$ $IK_{11} (IK_{111}, IK_{112}, \dots, IK_{11n_1});$ \dots

<p>С. 100–101.</p> <p>Формально ДІК визначається структурою вигляду $TIK = (I, \Omega)$, де $I = \{IK\}$ – множина ІК, $\Omega = I : I \rightarrow P$, де P – множина ребер, що відповідає матриці інцидентності.</p> <p>З кожним інтелектуальним компонентом ІКСА зв'язуватимемо його формальну об'єктну систему (ФОС), таким чином, ДІК буде пов'язуючим елементом між логічною і об'єктною структурами ІКСА.</p>  <p>Рис. 3.1. Топологія дерева інтелектуальних компонентів</p>	<p>С. 245–246.</p> <p>Формально ДІК определяется структурой вида $TIK = (I, \Omega)$, где $I = \{IK\}$ - множество ИК, $\Omega = I : I \rightarrow P$, где P - множество рёбер, что соответствует матрице инцидентности.</p> <p>С каждым интеллектуальным компонентом КИНС ППР будем связывать его формальную объектную систему, таким образом, ДІК явится связующим элементом между логической и объектной структурами КИНС ППР.</p>  <p>Рис. 4.1. Дерево интеллектуальных компонентов</p>
<p>С. 101.</p> <p>ДО СКЛАДУ даного інтелектуального компонента ІКі входять підлеглі йому інтелектуальні компоненти ІКіj, інтелектуальні агенти і інформаційні об'єкти, що мають продукційний інтелект. Тоді з кожною вершиною ДІК буде зв'язаний список вигляду $IK_i (IK_{i1}, IK_{i2}, \dots, IK_{im}; IA_{i1}, IA_{i2}, \dots, IA_{im}; O_{i1}, O_{i2}, \dots, O_{ik})$.</p> <p>З УРАХУВАННЯМ структури ІК структурно-логічна модель ІКСА визначається таким чином:</p> $T(IK, IA, IO) = (I, P, \Omega, A),$ <p>де P – множина зв'язків між ІК; $A = \{Ag, c, PA, \sigma Ag, PAO, \sigma AO, POB, \sigma OB\}$, де Ag – множина інтелектуальних агентів, які належать і-му ІК, $Ag = \{IA_1, \dots, IA_{im}\}$, Ob – множина інформаційних об'єктів, які належать і-му ІК $Ob = \{O_{i1}, \dots, O_{ik}\}$, σAg – відображення, яке визначає зв'язки IA, $\sigma Ag = Ag \times Ag \rightarrow PA$, PA – множина зв'язків IA, σAO – відображення, яке визначає зв'язки IA з IO $\sigma AO = Ag \times Ob \rightarrow PAO$, PAO – множина зв'язків IA з IO; σOB – відображення, яке визначає зв'язки на $\sigma OB = Ob \times Ob \rightarrow POB$, POB – множина зв'язків IO з іншим IO.</p>	<p>С. 247.</p> <p>В состав данного интеллектуального компонента ІКі будут входят подчиненные ему интеллектуальные компоненты ІКіj, интеллектуальные агенты и информационные объекты, обладающие продукционным интеллектом. Тогда с каждой вершиной ДІК будет связан список вида $IK_i (IK_{i1}, IK_{i2}, \dots, IK_{im}; IA_{i1}, IA_{i2}, \dots, IA_{im}; O_{i1}, O_{i2}, \dots, O_{ik})$.</p> <p>С учетом структуры ІК структурно-логическая модель КИНС ППР определяется следующим образом:</p> $T(IK, IA, IO) = (I, P, \Omega, A),$ <p>где P - множество связей между ИК; $A = \{Ag, c, PA, \sigma Ag, PAO, \sigma AO, POB, \sigma OB\}$, где Ag - множество интеллектуальных агентов, принадлежащих i-му ИК, $Ag = \{IA_1, \dots, IA_{im}\}$; Ob - множество информационных объектов, принадлежащих i-му ИК, $Ob = \{O_{i1}, \dots, O_{ik}\}$, σAg — отображение, определяющее связи IA, $\sigma Ag = Ag \times Ag \rightarrow PA$, PA — множество связей IA, σAO - отображение, определяющее связи IA и IO, $\sigma AO = Ag \times Ob \rightarrow PAO$, PAO - множество связей IA с IO; σOB - отображение, определяющее связи на Ob, $\sigma OB = Ob \times Ob \rightarrow POB$, POB – множество связей IO с IO.</p>
<p>С. 102.</p> <p>Логічна структура ІКСА в термінах інтелектуальних компонентів набуває вигляду, представленого на рис. 3.2. Логічна структура ІКСА в термінах структур даних може бути представлена деревом списків наступного виду (рис. 3.3). Така логічна організація дозволяє розглядати ІКСА з потрібним ступенем деталізації інтелектуальних компонентів, інтелектуальних агентів, об'єктів і відносин між ними [127].</p> <p>Розробка представленої вище логічної структури ІКСА дозволяє поставити питання про структуру і зміст баз знань інтелектуальних компонентів, який і обговорюється далі.</p>	<p>С. 247.</p> <p>Логическая структура КИНС ППР в терминах интеллектуальных компонентов приобретает вид, представленный на рис. 4.2. Логическая структура КИНС ППР в терминах структур данных может быть представлена деревом списков следующего вида (рис. 4.3). Такая логическая организация позволяет рассматривать КИНС ППР с нужной степенью детализации интеллектуальных компонентов, интеллектуальных агентов, объектов и отношений между ними [A19], [A36].</p> <p>Разработка представленной выше логической структуры КИНС ППР позволяет поставить вопрос о структуре и содержании баз знаний интеллектуальных компонентов, который и обсуждается в следующем разделе.</p>
<p>С. 102.</p>	<p>С. 250.</p>

<p>3.2. Архітектура бази знань інтелектуальної інформаційно-керуючої системи аеропорту</p> <p>Традиційно під базу знань розуміється ядро експертної системи, сукупність, знань наочної області, записана на машинний носій у формі, зрозумілій експертів і користувачеві. При розробці структури БЗ для інформаційної системи авіаприймства необхідно виходити із структури і характеру тієї інформації, яку БЗ повинна містити.</p>	<p>4. 2. Архитектура баз знаний корпоративных интеллектуальных систем поддержки принятия решений</p> <p>Традиционно под базой знаний понимается ядро экспертной системы, совокупность знаний предметной области, записанная на машинный носитель в форме, понятной эксперту и пользователю [125]. При разработке структуры БЗ необходимо исходить из структуры и характера той информации, которую БЗ должна содержать.</p>
<p>С. 103.</p>	<p>С. 248.</p>
 <p>Рис. 3.2. Структурно-логічна модель ІІКСА в термінах інтелектуальних компонентів</p>	 <p>Рис. 4.2. Структурно-логическая модель КИНС ППР в терминах интеллектуальных компонентов</p>
<p>С. 103.</p>	<p>С. 250.</p>
<p>Для побудови розподілених інформаційних систем існує низка стратегій розподілу даних і знань:</p> <ul style="list-style-type: none"> централізована стратегія (єдина копія розташовується на одному вузлі або компоненті системи управління); стратегія поділу (єдина копія БЗ розділяється на непересічні підмножини які розподілені по інтелектуальних компонентах системи); стратегія дублювання (існує декілька копій БЗ, таких, що кожен інтелектуальний компонент має в своєму розпорядженні повну копію всіх баз знань); змішана стратегія (підтримується декілька копій БЗ, розділених на підмножини так, що ІК може містити довільну підмножину БЗ) [128]. 	<p>При построении распределенных систем сложился ряд стратегий распределения данных и знаний:</p> <ul style="list-style-type: none"> централизованная стратегия (единственная копия-БЗ располагается на одном узле или компоненте системы управления); стратегия разделения (единственная, копия БЗ разделяется на непересекающиеся подмножества, которые распределены по интеллектуальным компонентам системы); стратегия дублирования (существует несколько копий БЗ, таких, что каждый интеллектуальный компонент располагает полной копией всех БЗ); смешанная стратегия (поддерживается несколько копий БЗ, разделенных на подмножества так, что ИК может содержать произвольное подмножество БЗ) [28], [125], [379].
<p>С. 104.</p>	<p>С. 249.</p>

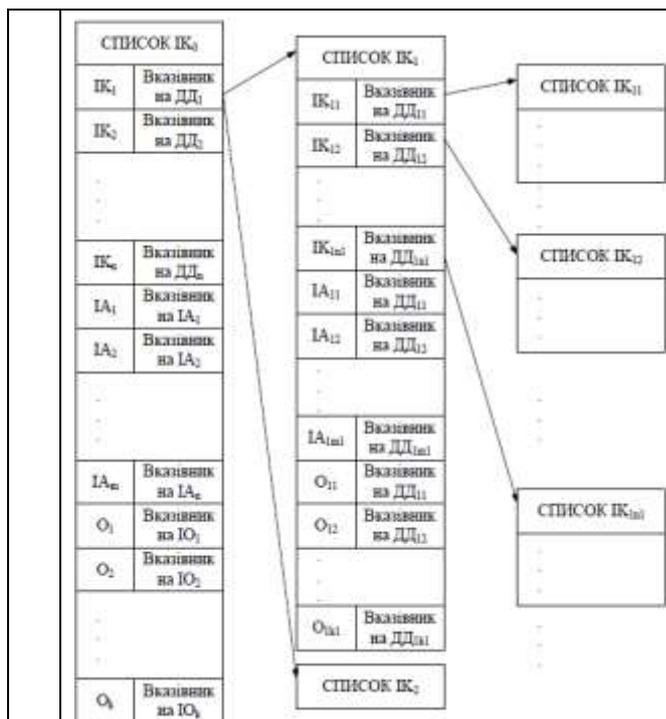


Рис. 3.3. Логічна структура ІКСА в термінах структур даних

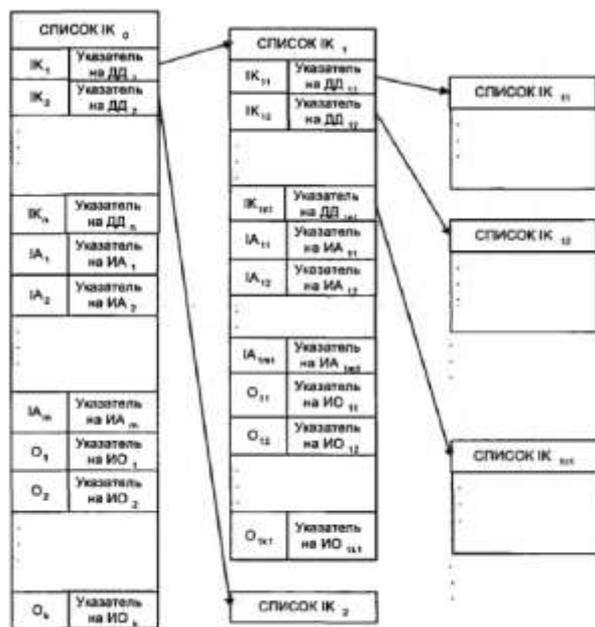


Рис. 4.3. Логическая структура КИНС ППР в терминах структур данных

С. 104–105.

Головною перевагою централізованої стратегії є простота організації БЗ, оскільки всі запити і транзакції здійснюються під контролем центрального вузла, яким, як правило, є сервер інформаційної системи. Недоліки даної стратегії виявляються в зростанні об'ємів баз знань і даних, що вимагає все більших обчислювальних потужностей і, відповідно, збільшення вартості таких серверів. З міркувань надійності доводиться виконувати резервування центральних баз знань і даних. Більшість існуючих інформаційних систем до цього часу реалізують централізовану стратегію, що виявляється в постійному зростанні вимог до обчислювальних ресурсів.

Стратегія поділу вимагає вирішення задачі розподілу знань по інтелектуальних компонентах системи, але дозволяє понизити комунікаційні витрати за рахунок розподілу інтелектуальних запитів по локальних вузлах системи.

Стратегія дублювання забезпечує значне підвищення надійності розподіленої системи, але вимагає істотних витрат пам'яті і узгодження множинних копій БЗ із забезпеченням виникаючих змін.

С. 105.

Змішана стратегія успадковує переваги і недоліки попередніх стратегій в тих пропорціях, в яких здійснюється симбіоз попередніх стратегій.

У контексті поставленого завдання БЗ інтелектуальних компонентів інформаційних систем аеропорту повинні відповідати низці вимог. По-перше, необхідно уніфікувати структуру БЗ інтелектуальних компонентів так, щоб вона не залежала від положення ІК в ДІК. Це забезпечить достатньо універсальний характер логічної моделі ІКСА, дозволить видозмінювати (нарошувати або скорочувати)

С. 250.

Главным достоинством централизованной стратегии является простота организации БЗ, так как все запросы и транзакции осуществляются под контролем центрального узла, каковым, как правило, является сервер корпоративной системы. Недостатки данной стратегии проявляются в росте объемов баз знаний и данных, что требует все больших вычислительных мощностей и, соответственно, увеличения стоимости таких серверов. По соображениям надежности приходится выполнять резервирование центральных баз знаний и данных. Большинство действующих корпоративных информационных систем до сих пор реализуют централизованную стратегию, что и проявляется в безудержном росте требований к вычислительным ресурсам.

Стратегия разделения требует решения задачи распределения знаний по интеллектуальным компонентам системы, но позволяет снизить коммуникационные затраты за счет распределения интеллектуальных запросов по локальным узлам системы.

Стратегия дублирования обеспечивает значительное повышение надежности распределенной системы, но требует существенных затрат памяти и согласования множественных копий БЗ с обеспечением возникающих изменений.

С. 251.

Смешанная стратегия наследует достоинства и недостатки предыдущих стратегий в тех пропорциях, в каких осуществляется симбиоз предыдущих стратегий.

В контексте решаемой автором задачи БЗ интеллектуальных компонентов должны отвечать ряду требований. Во-первых, необходимо унифицировать структуру БЗ интеллектуальных компонентов так, чтобы она не зависела от положения ІК в ДІК. Это обеспечит достаточно универсальный характер логической модели КИНС ППР, поз-

	<p>ДІК, не зачіпаючи внутрішню структуру БЗ. По-друге, БЗ інтелектуальних компонентів повинні забезпечувати зберігання тих моделей наочної області, в термінах яких проектується ІККА, тобто моделей логічної структури, моделей ІА, моделей ІО і моделей ФОС, що створюються на множині ІО.</p>	<p>волит видоизменять (наращивать или сокращать) ДИК, не затрагивая внутреннюю структуру БЗ. Во-вторых, БЗ интеллектуальных компонентов должны обеспечивать хранение тех моделей предметной области, в терминах которых проектируется КИНС ППР, т. е. моделей логической структуры, моделей ІА, моделей ІО и моделей ФОС, создаваемых на множествах ІО.</p>
	С. 105–106.	С. 251.
	<p>Аналіз сформульованих вище вимог до БЗ інтелектуальних компонентів ІККА примушує зупинитися на змішаній стратегії проектування БЗ. Це обумовлюється тим, що ІККА має ієрархічну організацію, де ІК верхніх рівнів повинні мати можливість управління і стеження за станом компонентів підлеглих рівнів. Оскільки окремі ІК можуть бути досить віддалені один від одного просторово, то велику частину знань вони повинні зберігати в своїх локальних БЗ (ЛБЗ), а вищим компонентам постачати знання, необхідні для підтримки цілісності і керованості ІККА. Разом з тим, системи аксіом і відносин, що встановлюються на ІК деякого рівня, повинні мати можливість «розповсюджуватися вниз», тобто бути відомими нижчим компонентам встановлюючи для них «правила поведінки». Тому в архітектурі ІККА передбачається можливе часткове дублювання БЗ з метою зниження комунікаційних витрат.</p>	<p>Анализ сформулированных выше требований к БЗ интеллектуальных компонентов КИНС ППР заставляет остановиться на смешанной стратегии проектирования БЗ. Это обусловливается тем, что КИНС ППР корпоративного типа имеют иерархическую организацию, где ИК верхних уровней должны иметь возможность управления и слежения за состоянием компонентов подчиненных уровней. Поскольку отдельные ИК могут быть весьма удалены друг от друга пространственно, то большую часть знаний они должны хранить в своих локальных БЗ (ЛБЗ), а вышестоящим компонентам поставлять знания, необходимые для поддержания целостности и управляемости КИНС ППР. Вместе с тем, системы аксиом и отношений, устанавливаемые на ИК некоторого уровня, должны иметь возможность «распространяться вниз», т. е. быть известными нижестоящим компонентам; устанавливая для них «правила поведения». Поэтому в архитектуре КИНС ППР предполагается возможное частичное дублирование БЗ с целью снижения коммуникационных затрат.</p>
	С. 106.	С. 251.
	<p>Таким чином, БЗ інтелектуального компонента можна представити у вигляді наступної конструкції: $KB_{IK_i} = (DD_i, M_{IA}, M_{IO}, KB_{FOS}),$ де DD_i – дерево декомпозиції інтелектуального компонента IK_i, M_{IA} – сукупність моделей IA, пов'язаних з даним IK, M_{IO} – сукупність моделей інформаційних об'єктів, пов'язаних з даним IK, KB_{FOS} — база знань формальної об'єктної системи даного IK [129].</p> <p>Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	<p>Таким образом, БЗ интеллектуального компонента можно представить в виде следующей конструкции: $KB_{IK_i} = (DD_i, M_{IA}, M_{IO}, KB_{FOS}),$ где DD_i - дерево декомпозиции интеллектуального компонента IK_i, M_{IA} – совокупность моделей IA, связанных с данным IK, M_{IO} - совокупность моделей информационных объектов, связанных с данным IK, KB_{FOS} - база знаний формальной объектной системы данного IK [A27].</p>
	С. 106.	С. 252.
	<p>Дерево декомпозиції інтелектуального компонента є структурою вигляду: $DD_i = IK_i (IK_{i1}, IK_{i2}, \dots, IK_{im}; IA_{i1}, IA_{i2}, \dots, IA_{im}; O_{i1}, O_{i2}, \dots, O_{ik}),$ тобто визначає зв'язок між логічною і об'єктною структурами ІККА на рівні даного ІК, в базі знань якого можна зберігати тільки посилання ІК нижчого рівня ($IK_{i1}, IK_{i2}, \dots, IK_{im}$). Таким чином, виникає однорідна ієрархічна структура дерев декомпозиції, яка закінчується на термінальних ІК, для яких $DD_i = IK_i (\emptyset; IA_{i1}, IA_{i2}, \dots, IA_{im}; O_{i1}, O_{i2}, \dots, O_{ik})$. Сукупність моделей ІО приймає вигляд $M_{IO} = (M_{i1}, M_{i2}, \dots, M_{im})$, де M_{ij} – модель інформаційного об'єкта O_{ij} в структурі бази знань ІК.</p>	<p>Дерево декомпозиции интеллектуального компонента представляет собой структуру вида $DD_i = IK_i (IK_{i1}, IK_{i2}, \dots, IK_{im}; IA_{i1}, IA_{i2}, \dots, IA_{im}; O_{i1}, O_{i2}, \dots, O_{ik}),$ т. е. определяет связь между логической и объектной структурами КИНС ППР на уровне данного ИК, в базе знаний которого можно хранить только ссылки на ИК нижестоящего уровня ($IK_{i1}, IK_{i2}, \dots, IK_{im}$). Таким образом, возникает однородная иерархическая структура деревьев декомпозиции, которая заканчивается на терминальных ИК, для которых $DD_i = IK_i (\emptyset; IA_{i1}, IA_{i2}, \dots, IA_{im}; O_{i1}, O_{i2}, \dots, O_{ik}).$ Совокупность моделей ИО принимает вид $M_{IO} = (M_{i1}, M_{i2}, \dots, M_{im}),$ где M_{ij} - модель информационного объекта O_{ij} в структуре базы знаний ИК.</p>

<p>С. 107.</p> <p>Згідно визначенням, введеним в розділі 2, інформаційний об'єкт є четвірка вигляду $O := \langle NO, \{A\}, \{O\}, \{F\} \rangle$, де NO – ім'я об'єкта; $\{A\}$ – множина атрибутів об'єкта (A_0, \dots, A_n), де A_i – i-й атрибут ІО; $\{O\}$ – множина об'єктів, які структурно входять до даного об'єкта, ($ONO_1, ONO_2, \dots, ONO_n$), де ONO_i i-й підпорядкований об'єкт, об'єкту з ім'ям NO; $\{F\}$ – множина функцій ІО.</p> <p>Перші три складові моделі ІО мають просту структуру і можуть бути представлені в базі знань ІК в табличній формі (рис. 3.4).</p> <p>Значно складніше задати функціональну модель ІО, формальне визначення якої дане в розділі 2. Щоб задати функції, потрібно перш за все визначитися з множиною пасивних інформаційних об'єктів (ПІО) $\{R\}$, що приймаються даним інформаційним об'єктом і множиною $\{T\}$ – передаваних даним інформаційним об'єктом ПІО.</p> <p>За визначенням, даним в розділі 2, ПІО є $PIO := \langle NO, \{A\}, \{O\}, \emptyset \rangle$.</p> <p>Виникає можливість побудови декількох моделей об'єктних систем залежно від трактування процесу обміну повідомленнями.</p>	<p>С. 252.</p> <p>Согласно определениям, введенным в разделе 2.2.2, информационный объект есть четверка вида $O := \langle NO, \{A\}, \{O\}, BM \rangle$,</p> <p>где NO - имя информационного объекта; $\{A\}$ - множество атрибутов объекта (A_0, A_2, \dots, A_n), где A_i - i-й атрибут ИО; $\{O\}$ — множество объектов, вложенных в данный объект ($ONO_1, ONO_2, \dots, ONO_i, \dots, ONO_m$), где ONO_i - i-вложенный объект объекта с именем NO, BM - модель поведения объекта.</p> <p>Первые три составляющие модели ИО имеют простую структуру и могут быть представлены в базе знаний ИК в табличной форме (рис. 4.4).</p> <p>Значительно сложнее обстоит дело с заданием модели поведения ИО, формальное определение которой дано в разделе 2.2.4. Чтобы задать модель поведения, нужно прежде всего определиться с множеством $\{R\}$ - принимаемых данным информационным объектом пассивных информационных объектов (ПИО) и множеством $\{T\}$ - передаваемых данным информационным объектом ПИО.</p> <p>По определению, данному в разделе 2.2.2, ПИО есть</p> <p style="text-align: center;">$PIO := \langle NO, \{A\}, \{O\}, \emptyset \rangle$.</p> <p>Возникает возможность построения нескольких моделей объектных систем в зависимости от трактовки процесса обмена сообщениями.</p>
<p>С. 107.</p> <p>Перша модель. У даній моделі передбачається, що всі елементи множини вкладених об'єктів $\{O\}$ є ПІО на всіх рівнях низхідної ієрархії, аж до останнього рівня структурних інформаційних об'єктів (СІО).</p> <p>Друга модель. У даній моделі передбачається, що в ПІО є елементи множини $\{O\}$, які не є ПІО. В цьому випадку у ПІО виникає прихована поведінка, що виражається в тому, що цей ПІО не може перетворювати свої атрибути, які приймають характер констант, але вкладені в нього ІО можуть змінювати свої атрибути, і оскільки їх модель поведінки не порожня, можуть у свою чергу передавати і приймати деякі ПІО.</p>	<p>С. 253.</p> <p>Первая модель. В данной модели предполагается, что все элементы множества вложенных объектов $\{O\}$ есть ПИО на всех уровнях нисходящей иерархии, вплоть до последнего уровня структурных информационных объектов (СИО).</p> <p>Вторая модель. В данной модели предполагается, что в ПИО есть элементы множества $\{O\}$, не являющиеся ПИО. В этом случае у ПИО возникает скрытое поведение, выражающееся в том, что этот ПИО не может преобразовывать свои атрибуты, которые принимают характер констант, но вложенные в него ИО могут изменять свои атрибуты, и поскольку их модель поведения не пуста, могут в свою очередь передавать и принимать некоторые ПИО.</p>
<p>С. 108.</p>	<p>С. 254.</p>

<table border="1" style="margin-bottom: 10px;"> <tr> <td style="width: 50px;">Ім'я ІО</td> <td style="width: 100px;">Вказівник на рядок символів</td> </tr> </table> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th colspan="3">Таблиця атрибутів</th> </tr> <tr> <th>Ім'я атрибута</th> <th>Множина визначення</th> <th>Значення атрибута</th> </tr> </thead> <tbody> <tr> <td>N_{A_1}</td> <td>S_{A_1}</td> <td>V_{A_1}</td> </tr> <tr> <td>N_{A_2}</td> <td>S_{A_2}</td> <td>V_{A_2}</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>N_{A_k}</td> <td>S_{A_k}</td> <td>V_{A_k}</td> </tr> </tbody> </table> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th colspan="2">Таблиця вкладених інформаційних об'єктів</th> </tr> </thead> <tbody> <tr> <td>O_{ij1}</td> <td>Вказівник на модель інформаційного об'єкта O_{ij1}</td> </tr> <tr> <td>O_{ij2}</td> <td>Вказівник на модель інформаційного об'єкта O_{ij2}</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>O_{ijf}</td> <td>Вказівник на модель інформаційного об'єкта O_{ijf}</td> </tr> </tbody> </table> <p style="text-align: center;">Рис. 3.4. Подання імені, атрибутів і вкладених об'єктів в базі знань інтелектуального компонента</p>	Ім'я ІО	Вказівник на рядок символів	Таблиця атрибутів			Ім'я атрибута	Множина визначення	Значення атрибута	N_{A_1}	S_{A_1}	V_{A_1}	N_{A_2}	S_{A_2}	V_{A_2}	N_{A_k}	S_{A_k}	V_{A_k}	Таблиця вкладених інформаційних об'єктів		O_{ij1}	Вказівник на модель інформаційного об'єкта O_{ij1}	O_{ij2}	Вказівник на модель інформаційного об'єкта O_{ij2}	O_{ijf}	Вказівник на модель інформаційного об'єкта O_{ijf}	<table border="1" style="margin-bottom: 10px;"> <tr> <td style="width: 50px;">Имя ИО</td> <td style="width: 100px;">Указатель на строку символов</td> </tr> </table> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th colspan="3">Таблиця атрибутів</th> </tr> <tr> <th>Имя атрибута</th> <th>Множество определения</th> <th>Значение атрибута</th> </tr> </thead> <tbody> <tr> <td>NA_1</td> <td>SA_1</td> <td>VA_1</td> </tr> <tr> <td>NA_2</td> <td>SA_2</td> <td>VA_2</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>NA_k</td> <td>SA_k</td> <td>VA_k</td> </tr> </tbody> </table> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th colspan="2">Таблиця вложенных информационных объектов</th> </tr> </thead> <tbody> <tr> <td>O_{ij1}</td> <td>Указатель на модель информационного объекта O_{ij1}</td> </tr> <tr> <td>O_{ij2}</td> <td>Указатель на модель информационного объекта O_{ij2}</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>O_{ijf}</td> <td>Указатель на модель информационного объекта O_{ijf}</td> </tr> </tbody> </table> <p style="text-align: center;">Рис. 4.4. Представление имени, атрибутов и вложенных объектов в базе знаний интеллектуального компонента</p>	Имя ИО	Указатель на строку символов	Таблиця атрибутів			Имя атрибута	Множество определения	Значение атрибута	NA_1	SA_1	VA_1	NA_2	SA_2	VA_2	NA_k	SA_k	VA_k	Таблиця вложенных информационных объектов		O_{ij1}	Указатель на модель информационного объекта O_{ij1}	O_{ij2}	Указатель на модель информационного объекта O_{ij2}	O_{ijf}	Указатель на модель информационного объекта O_{ijf}
Ім'я ІО	Вказівник на рядок символів																																																												
Таблиця атрибутів																																																													
Ім'я атрибута	Множина визначення	Значення атрибута																																																											
N_{A_1}	S_{A_1}	V_{A_1}																																																											
N_{A_2}	S_{A_2}	V_{A_2}																																																											
...																																																											
N_{A_k}	S_{A_k}	V_{A_k}																																																											
Таблиця вкладених інформаційних об'єктів																																																													
O_{ij1}	Вказівник на модель інформаційного об'єкта O_{ij1}																																																												
O_{ij2}	Вказівник на модель інформаційного об'єкта O_{ij2}																																																												
...	...																																																												
O_{ijf}	Вказівник на модель інформаційного об'єкта O_{ijf}																																																												
Имя ИО	Указатель на строку символов																																																												
Таблиця атрибутів																																																													
Имя атрибута	Множество определения	Значение атрибута																																																											
NA_1	SA_1	VA_1																																																											
NA_2	SA_2	VA_2																																																											
...																																																											
NA_k	SA_k	VA_k																																																											
Таблиця вложенных информационных объектов																																																													
O_{ij1}	Указатель на модель информационного объекта O_{ij1}																																																												
O_{ij2}	Указатель на модель информационного объекта O_{ij2}																																																												
...	...																																																												
O_{ijf}	Указатель на модель информационного объекта O_{ijf}																																																												
<p>С. 108–109.</p>	<p>С. 253.</p>																																																												
<p>Третя модель. Якщо ПІО, які приймають участь у процесах обміну повідомленнями, трактувати строго як СІО у яких $\{O\} = \emptyset$ і $\{F\} = \emptyset$, то виходить традиційна для програмування модель передачі параметрів [130–131].</p> <p>Аналізуючи першу модель, можна встановити, що в цьому випадку передається структура ПІО у вигляді дерева списків. Оскільки знання структури кореневого об'єкту в списку дозволяє звернутися і до всіх подальших елементів, то на рівні формальної моделі бази знань можна не обмежувати спільності ПІО, що приймаються, і елементи множини $\{A\}$ представляти таким чином (рис. 3.5). Такий підхід до множини $\{A\}$ не суперечить моделі поведінки ІО, викладеної в розділі 2, і схемам аксіом, правил виведення числення КМП. На рівні ж мови системного проектування ІКСА необхідно передбачити конструкції, що дозволяють конкретизувати множини $\{R\}$.</p> <p>Друга модель виявляється внутрішньо-суперечливою, оскільки ПІО, який за визначенням не володіє власною поведінкою ($\{F\} = \emptyset$), виявляється в змозі мати приховану внутрішню поведінку, якщо хоч би один з вкладених в нього ІО має не порожню модель функціонування ($\{F\} \neq \emptyset$).</p>	<p>Третья- модель. Если-ПІО,- участвующие-в-процессах обмена сообщениями, трактовать строго как СІО, у которых $\{O\} = \emptyset$ и $BM = \emptyset$, то получается традиционная для программирования парадигма передачи параметров, весьма хорошо изученная [380], [381].</p> <p>Анализируя первую модель, можно установить, что в этом случае передается структура ПІО в виде дерева списков. Поскольку знание структуры кореневого объекта в списке позволяет обратиться и ко всем последующим элементам, то на уровне формальной модели базы знаний можно не ограничивать общность принимаемых ПІО и элементов множества $\{L\}$ представлять следующим образом (рис. 4.5). Такой подход к множеству $\{R\}$ не противоречит модели поведения ИО, изложенной в разделе 2.2.4, и схемам аксиом правил вывода исчисления АТВМ. На уровне же языка системного проектирования КИНС ППР необходимо предусмотреть конструкции, позволяющие конкретизировать множество $\{R\}$.</p> <p>Вторая модель оказывается внутренне противоречивой, так как ПІО, который по определению не обладает собственным поведением ($BM = \emptyset$), оказывается в состоянии иметь скрытое внутреннее поведение, если хотя бы один из вложенных в него ИО имеет не пустую модель поведения ($BM \neq \emptyset$).</p>																																																												
<p>С. 109.</p>	<p>С. 255.</p>																																																												
<p>Пасивний ІО вигляду ПІО:=< NO,0,{O}, \emptyset> назовемо об'єктом-оболонкою (ПІОО – пасивний інформаційний об'єкт-оболонка). Такий об'єкт не має власної поведінки і власних атрибутів, тільки охоп-</p>	<p>Пассивный ИО вида ПІО:=< NO, 0, {O}, \emptyset> назовем объектом-оболочкой (ПІОО - пассивный информационный объект-оболочка). Такой объект не имеет собственного поведения и собственных</p>																																																												

лює вкладені об'єкти, хоча може мати приховану внутрішню поведінку. Тоді інформаційний об'єкт, всі вкладені ІО якого мають порожню функціональну модель ($\{F\} = \emptyset$), назовемо повністю пасивним ІО (ППІО). Тому необхідно накласти достатньо строге обмеження – всі ПІО, що належать множині $\{R\}$, повинні бути ППІО, що виражається наступною формулою: $\forall x((x \in \{R\}) \rightarrow (x = \text{ППІО}))$. Виконання цього обмеження можна перевірити шляхом обходу дерева ПІО як на стороні джерела, так і на стороні приймача.

атрибутов, тільки охоплює вкладені об'єкти, хоча може мати приховану внутрішню поведінку. Тогда информационный объект, все вложенные ИО которого имеют пустую модель поведения ($BM = \emptyset$), назовем полностью пассивным ИО (ППИО). Поэтому необходимо наложить достаточно строгое ограничение - все ПИО, принадлежащие множеству $\{R\}$, должны быть ППИО, что выражается следующей формулой: $\forall x((x \in \{R\}) \rightarrow (x = \text{ППИО}))$. Выполнение этого ограничения можно проверить путем обхода дерева ПИО как на стороне источника, так и на стороне приемника.

С. 110.

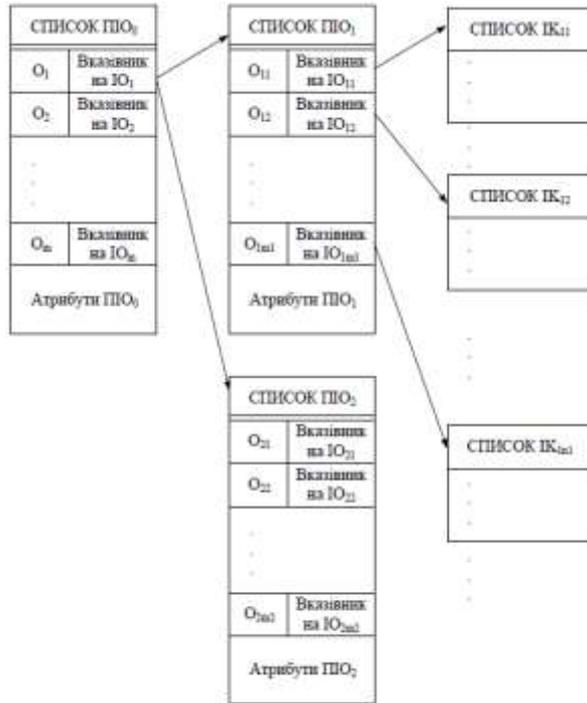


Рис. 3.5. Структура елемента множини повідомлень, що приймаються

С. 256.

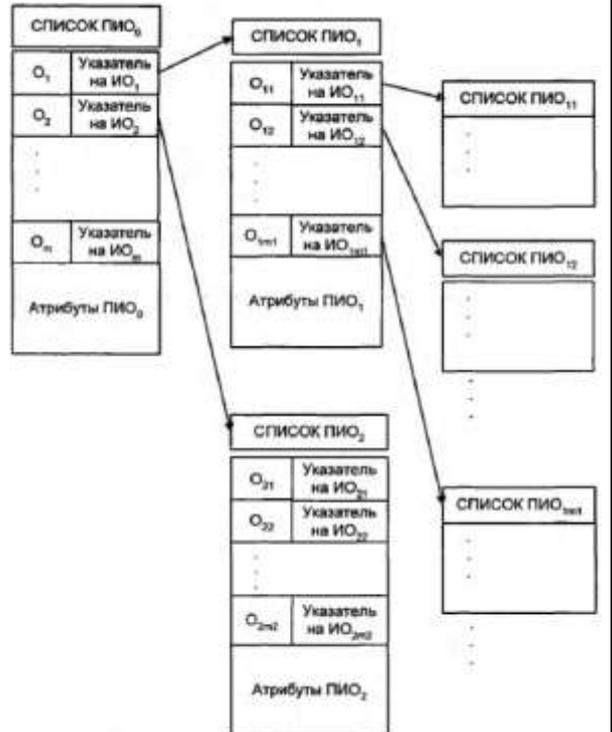


Рис. 4.5. Структура елемента множества принимаемых сообщений

С. 110–111.

Аналіз третьої моделі показує, що ця модель є окремим випадком першої моделі: $\forall x((x \in \{R\}) \rightarrow (x = \text{СІО}))$.
 На основі проведених міркувань слід зробити висновок про те, що перша модель обміну повідомленнями в ІКСА є найбільш адекватною змісту научної області і суті теорії інформаційних об'єктів, якими оперує інформаційна система аеропорту.

С. 255.

Анализ третьей модели показывает, что эта модель является частным случаем первой модели: $\forall x((x \in \{R\}) \rightarrow (x = \text{СИО}))$.
 На основе проведенных рассуждений следует сделать вывод о том, что первая модель обмена сообщениями в КИНС ППР является наиболее адекватной содержанию предметной области и сущности теории информационных объектов.

С. 111.

У ряді окремих випадків можна точніше визначити структуру елементів множини $\{R\}$, так в рамках третьої моделі виникають наступні варіанти:
 1. Множина $\{R\}$ кінцева і явно зліченна. Елементи цієї множини приймають вид записів: <ім'я_СІО; список_атрибутів; ім'я_атрибута, множина_визначення, значення...>.
 2. Точно задається тільки кількість передаваних атрибутів, а їх зміст визначається відправником і приймач самостійно декодує структуру атрибутів. Елементи множини $\{R\}$ приймають вид записів: <ім'я_СІО; список_атрибутів: <кількість_атрибутів>>.

С. 255.

В ряде частных случаев можно более точно определить структуру элементов множества $\{R\}$, так в рамках третьей модели возникают следующие варианты:
 1. Множество $\{R\}$ конечно и явно перечислимо. Элементы этого множества принимают вид записей:
 имя_СИО:
 список_атрибутов:
 имя_атрибута, множество_определения, значение...
 2. Точно задается только количество передаваемых атрибутов, а их содержание определяется отправителем, и приемник самостоятельно декодирует структуру атрибутов. Элементы множества

		{R} принимают вид записей: Имя_СИО: список_атрибутов: <число_атрибутов>.
С. 111.		С. 257.
<p>3. В цьому випадку не задається точна структура атрибутів, що приймаються, і їх кількість. Елементи множини {R} матимуть достатньо вільний формат в рамках узагальненої структури зв'язаних списків. Це можливо, коли заздалегідь невідомо, які атрибути зміняться і джерело повідомлення формує список атрибутів динамічно і достатньо складним чином.</p> <p>Оскільки елементи множини {T} можуть бути розглянуті аналогічно елементам множини {R}, то вважатимемо, що проведені міркування про властивості першої моделі обміну повідомленнями справедливі і для множини {T}.</p>	<p>3. В этом случае не задается точная структура принимаемых атрибутов и их количество. Элементы множества {R} будут иметь достаточно свободный формат в рамках обобщенной структуры связанных списков. Это возможно, когда неизвестно заранее, какие атрибуты изменятся, и источник сообщения формирует список атрибутов динамически и достаточно сложным образом.</p> <p>Поскольку элементы множества {T} могут быть рассмотрены аналогично элементам множества {R}, то будем считать, что проведенные рассуждения о свойствах первой модели обмена сообщениями справедливы и для множества {T}.</p>	
С. 111–112.		С. 257.
<p>В рамках обраної моделі у разі, коли елементи множин {R} і {T} є строго СИО, формат цих елементів і, відповідно, структуру повідомлень можна визначити через породжуючі контекстно-вільні граматики (КВГ). Для варіанта 3 породжуюча КВГ приймає вигляд:</p> <p><повідомлення := ім'я_СИО><список_атрибутов > <список_атрибутов > ::= <список_атрибутов > атрибут > <атрибут > <атрибут > ::= <ім'я_атрибутов > <множина_визначення > <значення >.</p> <p>Для варіанта 2 може бути складена наступна КВГ:</p> <p><повідомлення := ім'я_СИО><список_атрибутов > <список_атрибутов > ::= <кількість_атрибутов > <кількість_атрибутов > ::= <ціле_число_без_знака >.</p>	<p>В рамках выбранной модели в случае, когда элементы множеств {R} и {T} есть строго СИО, формат этих элементов и, соответственно, структуру сообщений можно определить через порождающие контекстно-свободные грамматики (КСГ).</p> <p>Для варианта 3 порождающая КСГ принимает вид</p> <p><сообщением := имя_СИО><список_атрибутов > <список_атрибутов > ::= список_атрибутов x атрибут > <атрибут > <атрибут > ::= имя_атрибутов > <множество_определения > <значение >.</p> <p>Для варианта 2 может быть составлена следующая КСГ:</p> <p><сообщением := имя_СИО><список_атрибутов > <список_атрибутов > ::= количество_атрибутов > <количество_атрибутов > ::= <целое_число_без_знака >.</p>	
С. 112.		С.257–258.
<p>У разі першого варіанта на КВГ повинні бути накладені жорсткі обмеження по кількості і структурі атрибутів. Отже, можна зробити висновок, що для розміщення елементів множин {R} і {T} в БЗ ІК буде достатньо запропонованих структур у формі дерев списків. Це дозволяє відмовитися від “жорстких” алгоритмічних схем і підтверджує спільність об'єктної моделі, запропонованої в розділі 2.</p> <p>Наступним елементом моделі функціонування ІО є множина допустимих предикатів {Pr}. Слідуючи [132] вважаємо, що предикат від n змінних, n-арний предикат на множині A є n-місна функція, визначена на A із значеннями в множині {істина, хибність}. Сукупність наборів (a1, a2, ..., an) елементів з A, для яких P(a1, a2, ..., an) – істина, називається n-арним відношенням, що відповідає предикату P. І навпаки, будь-якому n-арному відношенню R на A відповідає предикат P(x1, x2, ..., xn) на A такий, що</p> $P(a_1, a_2, \dots, a_n) = \begin{cases} \text{істина, якщо } (a_1, a_2, \dots, a_n) \in R \\ \text{хибність, якщо } (a_1, a_2, \dots, a_n) \notin R \end{cases}$	<p>В случае первого варианта на КСГ должны быть наложены жесткие ограничения по количеству и структуре атрибутов. Следовательно, можно сделать вывод, что для размещения элементов множеств {R} и {T} в БЗ ИК будет достаточно предложенных структур в форме деревьев списков. Это позволяет отказаться от «жестких» алгоритмических схем и подтверждает общность объектной модели, предложенной в разделе 2.2.</p> <p>Следующим элементом модели поведения ИО является множество допустимых предикатов {Pr}. Следуя [1], [382], полагаем, что предикат, от n переменных, n-арный предикат на множестве A есть n-местная функция, определенная на A со значениями в множестве {истина, ложь}. Совокупность наборов (a1, a2, ..., an) элементов из A, для которых P(a1, a2, ..., an) - истина, называется n-арным отношением, отвечающим предикату P. И наоборот, любому n-арному отношению R на A отвечает предикат P(x1, x2, ..., xn) на A такой, что</p> $P(a_1, a_2, \dots, a_n) = \begin{cases} \text{истина, если } (a_1, a_2, \dots, a_n) \in R \\ \text{ложь, если } (a_1, a_2, \dots, a_n) \notin R \end{cases}$	
С. 112–113.		С. 258.
<p>Предикат можна також розглядати як функцію, значеннями якої є вислови про n-ки об'єктів, що</p>	<p>Предикат можно также рассматривать как функцию, значениями которой являются высказы-</p>	

<p>представляють значення аргументів. Щоб задати n-місний предикат $P(x_1, x_2, \dots, x_n)$, слід вказати множини, D_1, D_2, \dots, D_n – області зміни предметних змінних x_1, x_2, \dots, x_n. З теоретико-множинної точки зору предикат визначається заданням підмножини M в декартовому добутку $D_1 \times D_2 \times \dots \times D_n$. При цьому $P(x_1, x_2, \dots, x_n)$ розуміється як вислів “впорядкований набір (a_1, a_2, \dots, a_n) належить M”.</p> <p>Синтаксично визначення n-місного предиката здійснюється вказівкою формули логіко-математичної мови, що містить n вільних змінних. Якщо P є n-місний предикативний символ, то серед синтаксичних правил утворення виразів формальної мови повинне бути правило: “якщо t_1, t_2, \dots, t_n – терми, то $P(t_1, t_2, \dots, t_n)$ – формула”. У ФОС над множиною IO роль такого правила виконує визначення з розділу 2. Виходить, що множина $\{Pr\}$ є список предикатів, допустимих для множини IO в даній ФОС. Областю зміни предметних змінних предикатів з множини $\{Pr\}$ буде множина IO, в окремому випадку – $\{Oop\}$ – множина можливих однопараметричних IO.</p>	<p>вания об n-ках объектов, представляющих значения аргументов. Чтобы задать n-местный предикат $P(x_1, x_2, \dots, x_n)$, следует указать множества D_1, D_2, D_n, - области изменения предметных переменных x_1, x_2, \dots, x_n. С теоретико-множественной точки зрения предикат определяется заданием подмножества M в декартовом произведении $D_1 \times D_2 \times \dots \times D_n$. При этом $P(x_1, x_2, \dots, x_n)$ понимается как высказывание «упорядоченный набор (a_1, a_2, \dots, a_n) принадлежит M».</p> <p>Синтаксически задание n-местного предиката осуществляется указанием формулы логико-математического языка, содержащей n свободных переменных. Если P есть n-местный предикатный символ, то среди синтаксических правил образования выражений формального языка должно быть правило: «если t_1, t_2, \dots, t_n - термы, то $P(t_1, t_2, \dots, t_n)$ - формула». В рассматриваемой автором ФОС над множеством IO роль такого правила выполняет определение 3 из раздела 2.2.2. Получается, что множество $\{Pr\}$ есть список предикатов, допустимых для множества IO в данной ФОС. Областью изменения предметных переменных предикатов из множества $\{Pr\}$ будет множество IO, в частном случае - $\{Oop\}$ - множество всевозможных однопараметрических IO.</p>																																																															
<p>С. 113.</p>	<p>С. 258.</p>																																																															
<p>Для маніпулювання з атрибутами IO використовуватимемо функцію визначення атрибуту $I(p, q)$, введenu в розділі 2, яка повертає однопараметричний IO, отже, можна побудувати формули в численні висловів, використовуючи цю функцію. Аналогічно можуть бути використані функції структурної потужності об'єкта P_w і первинної потужності P_p. Оскільки елементи множини $\{R\}$ є ППІО, то можна маніпулювати з їх атрибутами і потужностями, використовуючи функції I, P_w, P_p.</p> <p>Предикати множини $\{Pr\}$ слід визначити на рівні ФОС даного ІК. Множина $\{Pr\}$ може бути задана таблицею предикативних символів (табл. 3.1):</p>	<p>Для манипулирования с атрибутами IO будем использовать функцию извлечения атрибута $I(p, q)$, введенную в разделе 2.2.2, которая возвращает однопараметрический IO, следовательно, можно построить формулы в исчислении высказываний, используя эту функцию. Аналогично могут быть использованы функции структурной мощности объекта P_w и первичной мощности P_p. Поскольку элементы множества $\{R\}$ есть ППИО, то можно манипулировать с их атрибутами и мощностями, используя функции I, P_w, P_p.</p> <p>Предикаты множества $\{Pr\}$ следует определить на уровне ФОС данного ИК. Множество $\{Pr\}$ может быть задано таблицей предикатных символов (табл. 4.1).</p>																																																															
<p>С. 113–114.</p>	<p>С. 259.</p>																																																															
<p style="text-align: center;">Таблиця 3.1</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Ім'я предиката (відношення)</th> <th>Аристь предиката (відношення)</th> <th>Символ предиката (відношення)</th> </tr> </thead> <tbody> <tr> <td><ім'я_відношення></td> <td><ціле_число></td> <td><знак_відношення_у_формальній_системі></td> </tr> </tbody> </table> <p>Для встановлення правил виведення числення КМП у базі знань ІК використовуємо наступне табличне уявлення (табл. 3.2).</p> <p style="text-align: center;">Таблиця 3.2</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th colspan="2">Антецедент</th> <th colspan="2">Консеквент</th> </tr> <tr> <th>P_1</th> <th>P_2</th> <th>P_1^*</th> <th>P_2^*</th> </tr> </thead> <tbody> <tr> <td>P_2</td> <td></td> <td>P_2^*</td> <td></td> </tr> <tr> <td>P_4</td> <td></td> <td>P_4^*</td> <td></td> </tr> <tr> <td>P_5</td> <td></td> <td>P_5^*</td> <td></td> </tr> <tr> <td>P_6</td> <td></td> <td>P_6^*</td> <td></td> </tr> </tbody> </table>	Ім'я предиката (відношення)	Аристь предиката (відношення)	Символ предиката (відношення)	<ім'я_відношення>	<ціле_число>	<знак_відношення_у_формальній_системі>	Антецедент		Консеквент		P_1	P_2	P_1^*	P_2^*	P_2		P_2^*		P_4		P_4^*		P_5		P_5^*		P_6		P_6^*		<p style="text-align: center;">Таблиця 4.1</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>Ім'я предиката (отношения)</th> <th>Аристь предиката (отношения)</th> <th>Символ предиката (отношения)</th> </tr> </thead> <tbody> <tr> <td><имя_отношения></td> <td><целое_число></td> <td><знак, обозначающий отношение в формальной системе></td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> </tbody> </table> <p>Для задания правил вывода исчисления в базе знаний ИК используем следующее табличное представление (табл. 4.2).</p> <p style="text-align: center;">Таблиця 4.2</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th colspan="2">Антецедент</th> <th colspan="2">Консеквент</th> </tr> <tr> <th>P_1</th> <th>P_2</th> <th>P_1^*</th> <th>P_2^*</th> </tr> </thead> <tbody> <tr> <td>P_3</td> <td></td> <td>P_3^*</td> <td></td> </tr> <tr> <td>P_4</td> <td></td> <td>P_4^*</td> <td></td> </tr> <tr> <td>P_5</td> <td></td> <td>P_5^*</td> <td></td> </tr> <tr> <td>P_6</td> <td></td> <td>P_6^*</td> <td></td> </tr> </tbody> </table>	Ім'я предиката (отношения)	Аристь предиката (отношения)	Символ предиката (отношения)	<имя_отношения>	<целое_число>	<знак, обозначающий отношение в формальной системе>	Антецедент		Консеквент		P_1	P_2	P_1^*	P_2^*	P_3		P_3^*		P_4		P_4^*		P_5		P_5^*		P_6		P_6^*	
Ім'я предиката (відношення)	Аристь предиката (відношення)	Символ предиката (відношення)																																																														
<ім'я_відношення>	<ціле_число>	<знак_відношення_у_формальній_системі>																																																														
Антецедент		Консеквент																																																														
P_1	P_2	P_1^*	P_2^*																																																													
P_2		P_2^*																																																														
P_4		P_4^*																																																														
P_5		P_5^*																																																														
P_6		P_6^*																																																														
Ім'я предиката (отношения)	Аристь предиката (отношения)	Символ предиката (отношения)																																																														
<имя_отношения>	<целое_число>	<знак, обозначающий отношение в формальной системе>																																																														
...																																																														
Антецедент		Консеквент																																																														
P_1	P_2	P_1^*	P_2^*																																																													
P_3		P_3^*																																																														
P_4		P_4^*																																																														
P_5		P_5^*																																																														
P_6		P_6^*																																																														
<p>С. 114–115.</p>	<p>С. 259–260.</p>																																																															

<p>Тут П1 – поле мітки * для правил виводу, побудованих за схемами 7-10, П1* – поле мітки * для правил виводу, побудованих за схемами 2, 5, 6, 9, 10, П2 – посилання на список об'єкта Ri, тобто очікуваного поточного об'єкта з множини {R} У черзі вхідних об'єктів. Як було визначено раніше, це буде обов'язково ППІО. Список об'єкта Ri має вигляд, представлений на рис. 3.5. Поле П2* – номер поточної позиції черги вхідних ІО, прийняте повідомлення Ri видаляється з черги вхідних ІО. Значення поточного номера може формуватися по-різному залежно від того, як буде організована черга вхідних ІО. Поле П3 – номер поточного стану ІО, П3* – номер наступного стану. Поле П4 містить посилання на список атрибутів hA ДАНОГО ІО в поточному стані, а П4* є посиланням на структуру функціонального перетворення hA(Ri). Поле П5 – це посилання на поточну позицію черги вихідних ІО, а П5* – посилання на структуру функціонального породження Tj. Поле П6 – посилання на формулу Fj для правил, що породжуються за схемами 4, 6, 8, 10. Для правил, відповідних схемам 1–3, 5, 7, 9 – це порожнє посилання; Поле П6* – посилання на формулу Fj для правил, відповідних схемам 1, 3, 5, 7, 9, для правил же сформованих по схемах 4, 6, 8,10, – це порожнє посилання.</p>	<p>Здесь П1 - поле метки * для правил вывода, построенных по схемам аксиом 7-10, П1* — поле метки * для правил вывода, построенных по схемам аксиом 2, 5, 6, 9, 10, П2 - ссылка на список объекта Ri, т. е. ожидаемого текущего объекта из множества {R} в очереди входящих объектов. Как было определено ранее, это будет обязательно ППІО. Список объекта Ri имеет вид, представленный на рис. 4.5. Поле П2* - номер текущей позиции очереди входящих ІО, принятое сообщение Ri удаляется из очереди входящих ІО. Значение текущего номера может формироваться по-разному в зависимости от того, как будет организована очередь входящих ІО. Поле П3* – номер текущего состояния ІО, - номер следующего состояния. Поле П4 содержит ссылку на список атрибутов hA данного ІО в текущем состоянии, а П4* является ссылкой на структуру функционального преобразования hA(Ri). Поле П5 - это ссылка на текущую позицию очереди выходящих ІО, а П5* - ссылка на структуру функционального порождения Tj. Поле П6 - ссылка на формулу Fi для правил, порождаемых по схемам аксиом 4, 6, 8, 10. Для правил, соответствующих схемам аксиом 1-3, 5, 7, 9 - это пустая ссылка. Поле П6* - ссылка на формулу Fj для правил, соответствующих схемам аксиом 1-3, 5, 7, 9, для правил же сформированных по схемам аксиом 4, 6, 8, 10, - это пустая ссылка.</p>																
<p>С. 115.</p>	<p>С. 260.</p>																
<p>Перейдемо до розгляду структури функціонального перетворення hA(Ri) і структури функціонального породження Tj. Функціональне перетворення hA(Ri) здійснюється активним ІО над списком своїх атрибутів, який представляється в БЗ відповідною таблицею (рис. 3.4).</p> <p>На рівні формальної моделі числення ІО множини SAi не визначені, тому неможливо визначити функціональне перетворення hA(Ri) з точністю до кожної операції. Для вирішення цього питання з кожним атрибутом Ai зв'яжемо покажчик (посилання) на його функціональне перетворення h(Ai). Тобто задамо підстановку hA(Ri)=h(h(Ai)). ТАКИМ ЧИНОМ, структура функціонального перетворення hA(Ri) матиме вигляд таблиці (табл. 3.3).</p> <p>Слюняєв пише «ТАКИМ ЧИНОМ», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	<p>Перейдем к рассмотрению структуры функционального преобразования hA(Ri) и структуры функционального порождения Tj. Функциональное преобразование hA(Rj) осуществляется активным ІО над списком своих атрибутов, который представляется в БЗ соответствующей таблицей (рис. 4.4).</p> <p>На уровне формальной модели исчисления ІО множества SAi не определены, поэтому не представляется возможным определить функциональное преобразование hA(Ri) с точностью до каждой операции. Для решения этого вопроса с каждым атрибутом Ai свяжем указатель (ссылку) на его функциональное преобразование h(Ai), т. е. зададим подстановку hA(Ri) = h(h(Ai)). Таким образом, структура функционального преобразования hA(Ri) будет иметь вид таблицы (табл. 4.3).</p>																
<p>С. 115.</p>	<p>С. 260.</p>																
<p style="text-align: center;">Таблиця 3.3</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Номер атрибута</th> <th style="width: 50%;">Посилання на функціональне перетворення h(Ai)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">Вказивник 1</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> <tr> <td style="text-align: center;">n</td> <td style="text-align: center;">Вказивник n</td> </tr> </tbody> </table>	Номер атрибута	Посилання на функціональне перетворення h(Ai)	1	Вказивник 1	n	Вказивник n	<p style="text-align: center;">Таблиця 4.3</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Номер атрибута</th> <th style="width: 50%;">Ссылка на функциональное преобразование h(Ai)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">Указатель 1</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> <tr> <td style="text-align: center;">n</td> <td style="text-align: center;">Указатель n</td> </tr> </tbody> </table>	Номер атрибута	Ссылка на функциональное преобразование h(Ai)	1	Указатель 1	n	Указатель n
Номер атрибута	Посилання на функціональне перетворення h(Ai)																
1	Вказивник 1																
...	...																
n	Вказивник n																
Номер атрибута	Ссылка на функциональное преобразование h(Ai)																
1	Указатель 1																
...	...																
n	Указатель n																
<p>С. 115.</p>	<p>С. 261.</p>																
<p>Посилання на функціональні перетворення h(Ai) може вказувати на процедуру або функцію, що визначає значення атрибута Ai.</p> <p>Розглянемо структуру функціонального породження Tj. Як було встановлено раніше, елементи множини {T} є ППІО і представляються деревом списків (рис. 3.5). В процесі породження Tj необхід-</p>	<p>Ссылка на функциональное преобразование h(Ai) может указывать на процедуру или функцию, определяющую значение атрибута Ai.</p> <p>Рассмотрим структуру функционального порождения Tj. Как было установлено ранее, элементы множества {T} являются ППІО и представляются деревом списков (рис. 4.5). В процессе порождения</p>																

но визначити структуру передаваного ППІО, списки атрибутів і їх значення. Для цього з атрибутами об'єктів, що породжуються в T_j , також слід зв'язати покажчики на функціональні перетворення. Структура функціонального породження матиме вигляд, представлений на рис. 3.6.

T_j необхідно определить структуру передаваемого ППІО, списки атрибутов и их значения. Для этого с атрибутами объектов, порождаемых в T_j , также следует связать указатели на функциональные преобразования. Структура функционального порождения будет иметь вид, представленный на рис. 4.6.

С. 116.

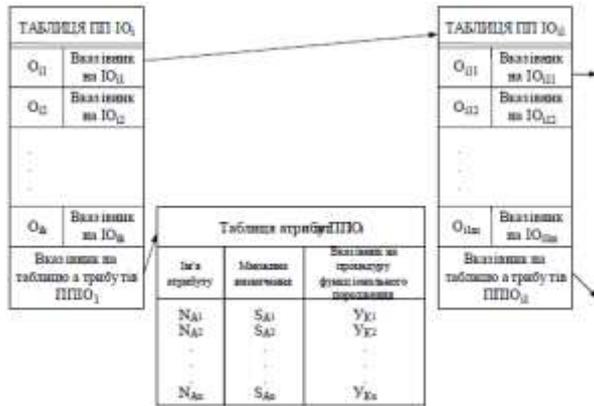


Рис. 3.6. Узагальнений вигляд структури породження

С. 262.

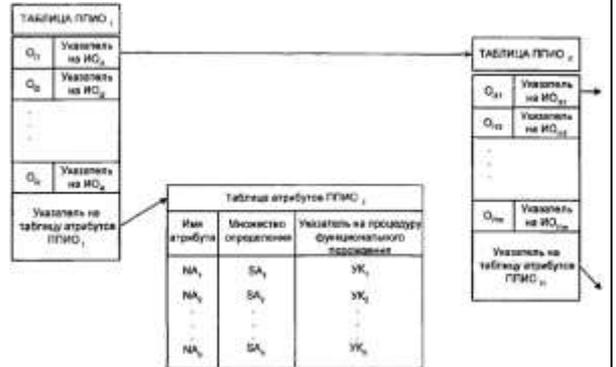


Рис. 4.6. Обобщенный вид структуры функционального порождения

С. 116.

Таким чином, визначені всі компоненти моделі ІО в БЗ ІК, узагальнений вид елементів МІО представляється структурою, зображеною на рис. 3.7. Розглянемо третю складову бази знань ІК – КВФОС. Досліджуємо структуру множини аксіом ФОС. Спираючись на визначення з розділу 2 слід представити множини аксіом ФОС як об'єднання трьох підмножин: $\{MA\} = (\{AG\} \cup \{ANG\} \cup \{ALI\})$, де $\{AG\}$ – множина глобальних аксіом, $\{ANG\}$ – множина нових глобальних аксіом, $\{ALI\}$ – множина локальних аксіом і-го інтелектуального компонента.

С. 261.

Таким образом, определены все компоненты модели ИО в БЗ ИК, обобщенный вид элементов МИО представляется структурой, изображенной на рис. 4.7. Рассмотрим третью составляющую базы знаний ИК – КВФОС. Исследуем структуру множества аксиом ФОС. Опираясь на определения раздела 2.2.2 и изложенное выше содержание раздела 4.1, следует представить множество аксиом ФОС как объединение трех подмножеств: $\{MA\} = (\{AG\} \cup \{ANG\} \cup \{ALI\})$, где $\{AG\}$ — множество глобальных аксиом, $\{ANG\}$ - множество новых глобальных аксиом, $\{ALI\}$ - множество локальных аксиом i-го интеллектуального компонента.

Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.

С. 117.

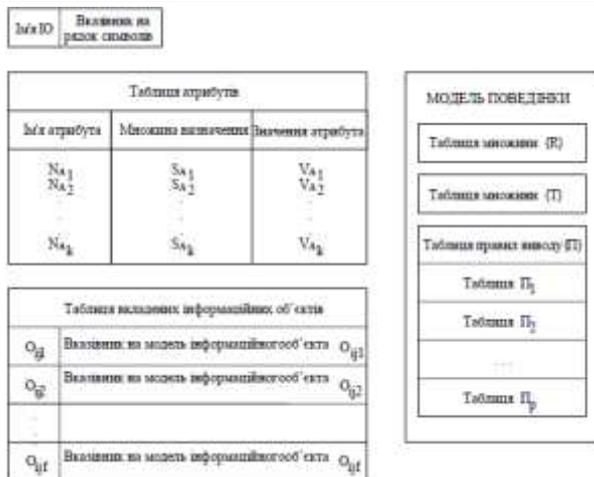


Рис. 3.7. Модель інформаційного об'єкта в базі знань інтелектуального компонента

С. 263. (На сторінці вказано 190)

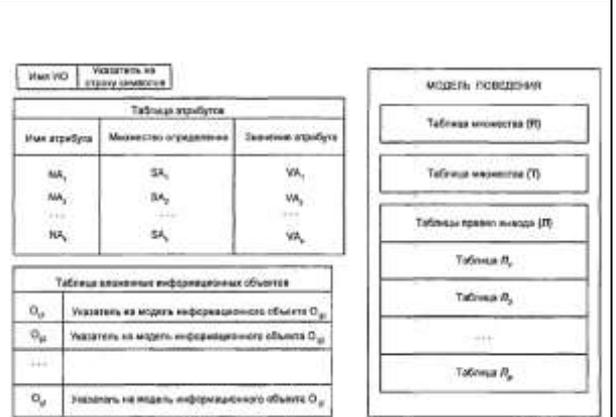


Рис. 4.7. Модель информационного объекта в базе знаний интеллектуального компонента

<p>С. 117.</p> <p>Множина {AG} – це множина аксіом априорно відомих ФОС всіх рівнів ієрархії, яке закладається на етапі проектування ІКСА (табл. 3.4).</p> <p>Список аксіом АО – це список аксіом, що виражають властивості симетричності, асиметричності, рефлексивності, антирефлексивності, транзитивності відносин. Аксіоми А.1–А.5, введені в розділі 2, по суті є схемами аксіом, так їх потрібно уточнювати для конкретного відношення, тому такі аксіоми слід помістити в окрему таблицю (табл. 3.5):</p>	<p>С. 261.</p> <p>Множество {AG} - это множество аксиом априорно известных ФОС всех уровней иерархии, которое закладывается на этапе проектирования КИНС ППР (табл. 4.4).</p> <p><...></p> <p>Список аксиом АО — это список аксиом, выражающих свойства симметричности, асимметричности, рефлексивности, антирефлексивности, транзитивности отношений. Аксиомы АО01 - АО05, введенные в разделе 2.2.2, по существу, являются схемами аксиом, так их требуется уточнять для конкретного отношения, поэтому такие аксиомы следует поместить в отдельную таблицу (табл. 4.5).</p>																																																
<p>С. 118.</p> <p style="text-align: center;">Таблиця 3.4.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Номер аксіом</th> <th style="width: 33%;">Область дії</th> <th style="width: 33%;">Символьний вираз</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><цисле_число_без_знака></td> <td style="text-align: center;"><вднoшення></td> <td style="text-align: center;"><символьний вираз></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;"><вказивник_на_символ_АО></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table>	Номер аксіом	Область дії	Символьний вираз	<цисле_число_без_знака>	<вднoшення>	<символьний вираз>	<вказивник_на_символ_АО>	<p>С. 261.</p> <p style="text-align: center;">Таблиця 4.4</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Номер аксіомы</th> <th style="width: 33%;">Область действия</th> <th style="width: 33%;">Символьное выражение</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><цисло_число_без_знака></td> <td style="text-align: center;"><отношение></td> <td style="text-align: center;"><символьное выражение></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;"> <указатель_на_список_АО></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table>	Номер аксіомы	Область действия	Символьное выражение	<цисло_число_без_знака>	<отношение>	<символьное выражение>	<указатель_на_список_АО>																								
Номер аксіом	Область дії	Символьний вираз																																															
<цисле_число_без_знака>	<вднoшення>	<символьний вираз>																																															
...	...	<вказивник_на_символ_АО>																																															
...																																															
Номер аксіомы	Область действия	Символьное выражение																																															
<цисло_число_без_знака>	<отношение>	<символьное выражение>																																															
...	...	<указатель_на_список_АО>																																															
...																																															
<p>С. 118.</p> <p style="text-align: center;">Таблиця 3.5</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Номер аксіом</th> <th style="width: 80%;">Символьний вираз</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>$\forall x \forall y (xRy \rightarrow yRx)$</td> </tr> <tr> <td style="text-align: center;">2</td> <td>$\forall x \forall y (xRy \rightarrow \neg yRx)$</td> </tr> <tr> <td style="text-align: center;">3</td> <td>$\forall x (xRx)$</td> </tr> <tr> <td style="text-align: center;">4</td> <td>$\forall x \neg (xRx)$</td> </tr> <tr> <td style="text-align: center;">5</td> <td>$\forall x \forall y \forall z ((xRy \& (yRz)) \rightarrow xRz)$</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table>	Номер аксіом	Символьний вираз	1	$\forall x \forall y (xRy \rightarrow yRx)$	2	$\forall x \forall y (xRy \rightarrow \neg yRx)$	3	$\forall x (xRx)$	4	$\forall x \neg (xRx)$	5	$\forall x \forall y \forall z ((xRy \& (yRz)) \rightarrow xRz)$	<p>С. 264.</p> <p style="text-align: center;">Таблиця 4.5</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Номер аксіомы</th> <th style="width: 85%;">Символьное выражение</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">01</td> <td>$\forall x \forall y (xRy \rightarrow yRx)$</td> </tr> <tr> <td style="text-align: center;">02</td> <td>$\forall x \forall y (xRy \rightarrow \neg (yRx))$</td> </tr> <tr> <td style="text-align: center;">03</td> <td>$\forall x (xRx)$</td> </tr> <tr> <td style="text-align: center;">04</td> <td>$\forall x \neg (xRx)$</td> </tr> <tr> <td style="text-align: center;">05</td> <td>$\forall x \forall y \forall z ((xRy) \& (yRz) \rightarrow xRz)$</td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table>	Номер аксіомы	Символьное выражение	01	$\forall x \forall y (xRy \rightarrow yRx)$	02	$\forall x \forall y (xRy \rightarrow \neg (yRx))$	03	$\forall x (xRx)$	04	$\forall x \neg (xRx)$	05	$\forall x \forall y \forall z ((xRy) \& (yRz) \rightarrow xRz)$																				
Номер аксіом	Символьний вираз																																																
1	$\forall x \forall y (xRy \rightarrow yRx)$																																																
2	$\forall x \forall y (xRy \rightarrow \neg yRx)$																																																
3	$\forall x (xRx)$																																																
4	$\forall x \neg (xRx)$																																																
5	$\forall x \forall y \forall z ((xRy \& (yRz)) \rightarrow xRz)$																																																
...	...																																																
Номер аксіомы	Символьное выражение																																																
01	$\forall x \forall y (xRy \rightarrow yRx)$																																																
02	$\forall x \forall y (xRy \rightarrow \neg (yRx))$																																																
03	$\forall x (xRx)$																																																
04	$\forall x \neg (xRx)$																																																
05	$\forall x \forall y \forall z ((xRy) \& (yRz) \rightarrow xRz)$																																																
...	...																																																
<p>С. 118–119.</p> <p>Множина {ANG} – це множина нових глобальних аксіом, що вводяться з новими відносинами на рівні інтелектуального компонента ІКО (табл. 3.6).</p> <p>На рівні конкретних ІК у ФОС можуть вводитися нові відносини, таким чином, виникають локальні множини {ALi} (табл. 3.7).</p> <p style="text-align: center;">Таблиця 3.6.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Номер аксіом</th> <th style="width: 33%;">Область дії</th> <th style="width: 33%;">Символьний вираз</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><цисле_число_без_знака></td> <td style="text-align: center;"><вднoшення></td> <td style="text-align: center;"><символьний вираз></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;"><вказивник_на_символ_АО></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table> <p style="text-align: center;">Таблиця 3.7.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Номер аксіом</th> <th style="width: 33%;">Область дії</th> <th style="width: 33%;">Символьний вираз</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><цисле_число_без_знака></td> <td style="text-align: center;"><вднoшення></td> <td style="text-align: center;"><символьний вираз></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;"><розповсюдження></td> <td style="text-align: center;"><вказивник_на_список_АО></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table> <p>Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	Номер аксіом	Область дії	Символьний вираз	<цисле_число_без_знака>	<вднoшення>	<символьний вираз>	<вказивник_на_символ_АО>	Номер аксіом	Область дії	Символьний вираз	<цисле_число_без_знака>	<вднoшення>	<символьний вираз>	...	<розповсюдження>	<вказивник_на_список_АО>	<p>С. 264.</p> <p>Множество {ANG} — это множество новых глобальных аксиом, вводимых с новыми отношениями на уровне интеллектуального компонента ІКО (табл. 4.6).</p> <p style="text-align: center;">Таблиця 4.6</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Номер аксіомы</th> <th style="width: 33%;">Область действия</th> <th style="width: 33%;">Символьное выражение</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><цисло_число_без_знака></td> <td style="text-align: center;"><отношение></td> <td style="text-align: center;"><символьное выражение></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;"> <указатель_на_список_АО></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table> <p>На уровне конкретных ІК в ФОС могут вводиться новые отношения, таким образом, возникают локальные множества {ALi} (табл. 4.7).</p> <p style="text-align: center;">Таблиця 4.7</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Номер аксіомы</th> <th style="width: 33%;">Область действия</th> <th style="width: 33%;">Символьное выражение</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><цисло_число_без_знака></td> <td style="text-align: center;"><отношение></td> <td style="text-align: center;"><символьное выражение></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;"><распространение></td> <td style="text-align: center;"> <указатель_на_список_АО></td> </tr> <tr> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> <td style="text-align: center;">...</td> </tr> </tbody> </table>	Номер аксіомы	Область действия	Символьное выражение	<цисло_число_без_знака>	<отношение>	<символьное выражение>	<указатель_на_список_АО>	Номер аксіомы	Область действия	Символьное выражение	<цисло_число_без_знака>	<отношение>	<символьное выражение>	...	<распространение>	<указатель_на_список_АО>
Номер аксіом	Область дії	Символьний вираз																																															
<цисле_число_без_знака>	<вднoшення>	<символьний вираз>																																															
...	...	<вказивник_на_символ_АО>																																															
...																																															
Номер аксіом	Область дії	Символьний вираз																																															
<цисле_число_без_знака>	<вднoшення>	<символьний вираз>																																															
...	<розповсюдження>	<вказивник_на_список_АО>																																															
...																																															
Номер аксіомы	Область действия	Символьное выражение																																															
<цисло_число_без_знака>	<отношение>	<символьное выражение>																																															
...	...	<указатель_на_список_АО>																																															
...																																															
Номер аксіомы	Область действия	Символьное выражение																																															
<цисло_число_без_знака>	<отношение>	<символьное выражение>																																															
...	<распространение>	<указатель_на_список_АО>																																															
...																																															
<p>С. 119.</p> <p>Визначимо область дії локальних аксіом. Для термінальних ІК аксіоми множини {ALi} можуть діяти тільки на рівні даного ІК, для нетермінальних можуть бути двох видів: локалізовані (що діють тільки у ФОС даного ІК); успадковані вниз (що діють на всі ФОС ІК, наступних вниз за ієрархією за да-</p>	<p>С. 264–265.</p> <p>Определим область действия локальных аксиом. Для терминальных ІК аксиомы множества {ALi} могут действовать только на уровне данного ІК, для нетерминальных могут быть двух видов: локализованные (действующие только в ФОС данного ІК); наследуемые вниз (действующие на все ФОС</p>																																																

<p>ним ІК, але що не діють на інші гілки дерева ІК). Тому ознака розповсюдження визначається наступною граматику:</p> <p>< розповсюдження >::=< локальне > < наслідуваниє_вниз > < локальне >::= L < наслідуваниє_вниз >::= D</p> <p>Множина {AG} закладається при проектуванні ІКСА і на рівні ІК0 повинен зберігатися еталонний зразок, а його копії необхідно зберігати у всіх останніх ІК. Множина {ANG} також повинна бути відома всім ІК, тому повинен існувати еталон на рівні ІК0 і його копії у всіх ІК. Ця множина може достатньо динамічно змінюватися, отже, на стадії функціонування ІКСА потрібна процедура синхронізації з еталоном ІК0, що ініціюється. Множина локальних аксіом {ALi} також може бути поширюваною, тому, якщо множина {ALi} є успадкованою вниз, то для всіх ІК піддерева ІК, слід зберігати копії цієї множини і синхронізувати їх у разі зміни.</p>	<p>ІК, следующих вниз по иерархии за данным ІК, но не действующие на другие ветви дерева ІК). Поэтому признак распространения определяется следующей грамматикой:</p> <p>< распространение >::=< локальное > < наследуемое_вниз > < локальное >::= L < наследуемое_вниз >::= D</p> <p>Множество {AG} закладывается при проектировании КИНС ППР и на уровне ІК0 должен храниться эталонный образец, а его копии необходимо хранить во всех остальных ІК. Множество {ANG} также должно быть известно всем ІК, поэтому должен существовать эталон на уровне ІК0 и его копии у всех ІК. Это множество может достаточно динамично изменяться, следовательно, на стадии функционирования КИНС ППР требуется процедура синхронизации с эталоном, иницируемая ІК0. Множества локальных аксиом {ALi} также могут быть распространяемыми, поэтому, если множество {ALi} является наследуемым вниз, то для всех ІК поддерева ІКі следует хранить копии этого множества и синхронизировать их в случае изменения.</p>																
<p>С. 119–120.</p>	<p>С. 265.</p>																
<p>Наступним компонентом КВФОС є відносини між ІО, які необхідно зберігати в базі фактів. Вважатимемо, що ІК зберігає інформацію про відносини ІО з ІК. База фактів зберігається на рівні даного ІК і містить пропозиції першого порядку мови а алгебраїчної системи Ф відповідно до визначень, даних в розділі 2 (визначення б) (табл. 3.8)</p> <p style="text-align: right;">Таблиця 3.8</p> <table border="1" data-bbox="284 1115 866 1240"> <thead> <tr> <th>Номер факта</th> <th>Вказівник на символічний вираз</th> </tr> </thead> <tbody> <tr> <td><число_число_без_знака></td> <td>Вказівник 1</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td></td> <td>Вказівник n</td> </tr> </tbody> </table>	Номер факта	Вказівник на символічний вираз	<число_число_без_знака>	Вказівник 1		Вказівник n	<p>Следующим компонентом КВФОС являются отношения между ІО, которые необходимо хранить в базе фактов. Будем полагать, что ІКі хранит информацию об отношениях ІО из ІКі. База фактов хранится на уровне данного ІК и содержит предложения первого порядка языка а алгебраической системы Ф в соответствии с определениями, данными в разделе 2.2.2 (определение б) (табл. 4.8).</p> <p style="text-align: right;">Таблиця 4.8</p> <table border="1" data-bbox="895 1144 1481 1270"> <thead> <tr> <th>Номер факта</th> <th>Указатель на символическое выражение</th> </tr> </thead> <tbody> <tr> <td><число_число_без_знака></td> <td>Указатель 1</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td></td> <td>Указатель n</td> </tr> </tbody> </table>	Номер факта	Указатель на символическое выражение	<число_число_без_знака>	Указатель 1		Указатель n
Номер факта	Вказівник на символічний вираз																
<число_число_без_знака>	Вказівник 1																
...	...																
	Вказівник n																
Номер факта	Указатель на символическое выражение																
<число_число_без_знака>	Указатель 1																
...	...																
	Указатель n																
<p>С. 120.</p>	<p>С. 265.</p>																
<p>Таблиця предикативних символів також включена в КВФОС оскільки допустимі предикати будуть загальними для всієї підмножини ІО, що входять в ІКі.</p> <p>Таким чином, розроблена архітектура баз знань інтелектуального компонента ІКСА, загальний вид якої представлена на рис. 3.8.</p> <p>Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	<p>Таблиця предикатных символов также включена в КВФОС, так как допустимые предикаты будут являться общими для всего подмножества ІО, входящих в ІКі.</p> <p>Таким образом, разработана архитектура баз знаний интеллектуального компонента КИНС ППР, общий вид которой представлена на рис. 4.8.</p>																
<p>С. 120.</p>	<p>С. 267.</p>																
<p>3.3. Алгоритми логічного виводу в архітектурі інтелектуальної інформаційно-керуючої системи аеропорту</p> <p>Вибір методу пошуку рішень і реалізації механізмів виводу визначається цілим рядом чинників: специфікою області застосування, розмірами простору пошуку, рівнем визначеності і надійності знань і даних, динамікою змін, що відбуваються в наочній області, моделями представлення знань, обчислювальними ресурсами, які можуть бути застосовані в інтелектуальній системі.</p>	<p>4. 3. Алгоритмы логиче ского вывода в архитектуре КИНС ППР</p> <p>Выбор метода поиска решений и реализации механизмов вывода определяется целым рядом факторов: спецификой предметной области, размерами пространства: поиска, уровнем определенности и надежности знаний и данных, динамикой происходящих в предметной области изменений, моделями представления знаний, вычислительными ресурсами, которые могут быть применены в интеллектуальной системе.</p>																

Оскільки розробляється аксіоматизована формальна система Φ над множиною ІО, то повинні застосовуватися дедуктивні методи пошуку рішень.

Поскольку разрабатывается аксиоматизируемая формальная система Φ над множеством ІО; то должны применяться дедуктивные методы поиска решений.

С. 121.



Рис. 3.8. Архітектура бази знань інтелектуального компонента в ІКСА

С. 266.

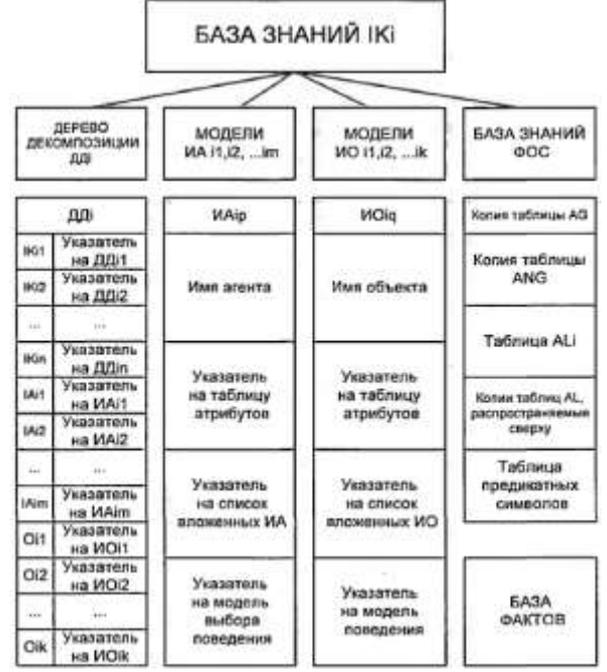


Рис. 4.8. Архитектура базы знаний интеллектуального компонента в КИНС ППР

С. 121–122.

У дедуктивних системах проблема, що підлягає вирішенню формулюється у вигляді тверджень деякої формальної мови, а мета – у вигляді твердження, справедливості якого необхідно довести або спростувати на підставі аксіом і правил виведення формальної системи. Проблема пошуку рішення зводиться до доведення теореми, що встановлює логічний вихід цільової формули з формул-посилок.

С. 267.

В дедуктивних системах подлежащая решению проблема формулируется в виде утверждений некоторого формального языка, а цель - в виде утверждения, справедливость которого необходимо доказать или опровергнуть на основании аксиом и правил вывода формальной системы. Проблема поиска решения сводится к доказательству теоремы, устанавливающей логическое следование целевой формулы из формул-посылок.

С. 122.

Дедуктивні методи успішно використовуються при побудові глобальних баз знань [133 – 140]. Глобальні БЗ містять десятки і сотні тисяч логічних формул і успішно функціонують.

Вважатимемо, що окрема ФОС є системою гільбертовського типу (Н-системою). Відповідно до [141] вважаємо, що схеми аксіом для Н – це все тавтології, тобто загальнозначущі формули в мові а першого порядку; всі аксіоми рівності; всі формули вигляду $(\forall v\varphi(v)) \rightarrow \varphi(t), \varphi(t) \rightarrow \exists v\varphi(v)$.

С. 269.

Приверженность автора диссертационной работы методам дедуктивного вывода решений подтверждается также и тем, что такие методы успешно используются при построении глобальных баз знаний (Very Large Knowledge Base - VLKB), таких как Cysc (Cyscorp) [389], HPKB (High Performance Knowledge Base), создаваемой Стэнфордской лабораторией KSL (Knowledge System Laboratory) [390], SRI [391]. Глобальные БЗ содержат десятки и сотни тысяч логических формул и, тем не менее, успешно функционируют.

Мы будем полагать, что отдельная ФОС является системой гильбертовского типа (Н-системой). В соответствии с [258] считаем, что схемы аксиом для Н - это все тавтологии, т. е. общезначимые формулы в языке а первого порядка; все аксиомы равенства; все формулы вида $(\forall v\varphi(v)) \rightarrow \varphi(t), \varphi(t) \rightarrow \exists v\varphi(v)$.

С. 122.

Правила вывода для Н: (Modus ponens). З $(\varphi \rightarrow \psi)$ і φ слідує ψ .
(Правила узагальнення). Якщо змінна v не входить вільно в φ , то із $\varphi \rightarrow \psi(v)$ слідує

С. 269.

Правила вывода для Н: (Modus ponens). Из $(\varphi \rightarrow \psi)$ и φ следует ψ .
(Правила обобщения). Если переменная v не

<p>$\varphi \rightarrow \forall u\varphi(u)$, із $\psi(v) \rightarrow \varphi$ слідує $\exists u\varphi(u) \rightarrow \varphi$.</p> <p>Для системи Н справедлива теорема Геделя про повноту, тобто кожна загальнозначуща формула виводиться.</p> <p>У тих ФОС, що розглядаються ми інтерпретуємо числення предикатів на наочній області ІО, що, природно, не впливає на загальнозначимість Н-системи. Припускаємо, що для даної ФОС, множини аксіом $\{AG\}$, $\{ANG\}$, $\{AL \dots\}$ з копіями утворює тавтологію, а база фактів дає нам формули $(\forall v\varphi(v)) \rightarrow \varphi(t), \varphi(t) \rightarrow \exists v\varphi(v)$.</p> <p>Алгоритми дедуктивного виводу в логічній структурі ІІКСА розробляємо, спираючись на інтерполяційну теорему Крейга і поняття графа перетинів. У роботах Крейга було показано, що якщо α і β – формули в численні предикатів і $\alpha \vdash \beta$, то існує формула, що включає тільки загальні символи для α і β такі що $\alpha \vdash \gamma$ і $\gamma \vdash \beta$ [142–143].</p>	<p>входить свободно в φ, то из $\varphi \rightarrow \psi(v)$ следует $\varphi \rightarrow \forall u\psi(u)$, из $\psi(v) \rightarrow \varphi$ следует $\exists u\psi(u) \rightarrow \varphi$.</p> <p>Для системы Н справедлива теорема Геделя о полноте, т. е. каждая общезначимая формула выводима.</p> <p>В рассматриваемых нами ФОС мы интерпретируем исчисление предикатов на предметной области ІО, что, естественно, не влияет на общезначимость Н-системы. Предполагаем, что для данной ФОС, множества аксиом $\{AG\}$, $\{ANG\}$, $\{AL \dots\}$ с копиями образуют тавтологии, а база фактов дает нам формулы вида $(\forall v\varphi(v)) \rightarrow \varphi(t), \varphi(t) \rightarrow \exists v\varphi(v)$.</p> <p>Алгоритмы дедуктивного вывода в логической структуре КИНС ППР мы разрабатываем, опираясь на интерполяционную теорему Крейга (Craig W.) и понятие графа пересечений. В работах Крейга было показано, что если α и β - формулы в исчислении предикатов и $\alpha \vdash \beta$, то существует формула γ, включающая в себя только общие символы для α и β, такие что $\alpha \vdash \gamma$ и $\gamma \vdash \beta$ [392], [393].</p>
<p>С. 122–123.</p>	<p>С. 270.</p>
<p>Для бази фактів кожного інтелектуального компонента – BFi позначимо через $J(BFi)$ сигнатуру, що включає множину нелогічних символів. Відповідно до [144] визначимо граф перетинів $G = (V, E, W)$, де множина $V = \{1, \dots, n\}$ – є множина вершин, відповідних базам фактів BFi, дві вершини i та j зв'язуються дугою, якщо $J(BFi)$ і $J(BFj)$ мають загальні символи $(J(BFi) \cap J(BFj) \neq \emptyset)$. Множина E – це множина дуг $E = \{(i, j) \mid J(BFi) \cap J(BFj) \neq \emptyset\}$.</p> <p>З дугами зв'язуються мітки, що належать множині $W = \{Wk(i, j) = J(BFi) \cap J(BFj)\}$, тобто які містять множини символів, за якими зв'язані бази фактів.</p>	<p>Для бази фактів кожного інтелектуального компонента - BFi, обозначим через $J(BFi)$ сигнатуру, включающую множество нелогических символов. В соответствии с [394] определим граф пересечений $G = (V, E, W)$, где множество $V = \{1, \dots, n\}$ - есть множество вершин, соответствующих базам фактов BFi, две вершины i и j связываются дугой, если $J(BFi)$ и $J(BFj)$ имеют общие символы $(J(BFi) \cap J(BFj) \neq \emptyset)$. Множество E - это множество дуг $E = \{(i, j) \mid J(BFi) \cap J(BFj) \neq \emptyset\}$.</p> <p>С дугами связываются метки, принадлежащие множеству $W = \{Wk(i, j) = J(BFi) \cap J(BFj)\}$, т. е. содержащие множества символов, по которым связаны базы фактов.</p>
<p>С. 123.</p>	<p>С. 270.</p>
<p>Для випадку дедуктивної теорії з розділеними базами знань відомий алгоритм виведення MP (Message-Passing) [145], який полягає в наступному.</p> <p>Початкові дані: множина баз знань $\{BFi\}_{i \leq n}$, які складають опис наочної області (теорію); граф перетинів $G = (V, E, W)$, що описує взаємозв'язок між базами; Q – цільова формула з сигнатурою $J(Q) \sqsubseteq J\{BFk\}$.</p> <p>Крок 1. Хай $dist(i, j) (i, j \in V)$ – довжина найкоротшого шляху між i, j у графі G. Хай $i \sqsubseteq j$ тільки якщо $dist(i, k) < dist(j, k)$. Тут \sqsubseteq є відношення строгого часткового порядку на множині V.</p> <p>Крок 2. Паралельно виконується послідовний пошук для кожної бази $BFi, i \leq n$.</p> <p>Крок 3. Для кожної пари $(i, j) \in E$ такої, що $i \sqsubseteq j$, якщо вдається довести $BFj - \varphi$ і $J(\varphi) \sqsubseteq J(W(i, j))$, то формула φ додається до BFi. Змістовно це означає рух від вершин в порядку зменшення відстані $dist(i, j)$ до вершини BFk додаванням в простір пошуку формул φ.</p>	<p>Для случая дедуктивной теории с разделенными базами знаний известен алгоритм вывода MP (forward message-passing algorithm) [395], который заключается в следующем.</p> <p>Исходные данные: множество баз знаний $\{BFi\}_{i \leq n}$, составляющих описание предметной области (теорию); граф пересечений $G = (V, E, W)$, описывающий взаимосвязь между базами; Q - целевая формула с сигнатурой $J(Q) \sqsubseteq J\{BFk\}$.</p> <p>Шаг 1. Пусть $dist(i, j) (i, j \in V)$ - длина кратчайшего пути между i, j в графе G. Пусть $i \sqsubseteq j$ только если $dist(i, k) < dist(j, k)$. Здесь \sqsubseteq есть отношение строгого частичного порядка на множестве V.</p> <p>Шаг 2. Параллельно выполняется последовательный поиск для каждой базы $BFi, i \leq n$.</p> <p>Шаг 3. Для каждой пары $(i, j) \in E$ такой, что $i \sqsubseteq j$, если удастся доказать $BFj - \varphi$ и $J(\varphi) \sqsubseteq J(W(i, j))$, то формула φ добавляется к BFi. Содержательно это означает движение от вершин в порядке убывания расстояния $dist(i, j)$ к вершине BFk с добавлением в пространство поиска формул φ.</p>
<p>С. 123–124.</p>	<p>С. 270–271.</p>
<p>Крок 4. Якщо вдається довести формулу Q на базі BFk то видається повідомлення "Мета досягнута", інакше пошук завершується безрезультатно.</p> <p>Алгоритм MP не дає можливості здійснити виведення цільової формули Q, такої, що $J(Q) \sqsubseteq J\{BFk\}$, тобто сигнатура цільової формули не належить цілком сигнатурі БЗ одного ІК. Для ІІКСА пошук таких формул цілком можливий, оскільки бази</p>	<p>Шаг 4. Если удается доказать формулу Q на базе BFk, то выдается сообщение «Цель достигнута», в противном случае поиск завершается безрезультатно.</p> <p>Алгоритм MP не дает возможности осуществить вывод целевой формулы Q, такой, что $J(Q) \sqsubseteq J\{BFk\}$, т.е. сигнатура целевой формулы не принадлежит целиком сигнатуре БЗ одного ІК. Для</p>

	<p>знань ІК зберігають знання певного рівня ієрархії і нові знання (в даному випадку складна цільова формула Q) можуть виникати в процесі сумісного виводу по декількох БЗ різних ІК. Тому пропонується алгоритм MPQ, що розвиває і удосконалює алгоритм MP, стосовно баз знань ІКСА.</p>
<p>С. 124.</p>	<p>С. 271.</p>
<p>Позначимо як T_{IK_i} – піддерево дерева інтелектуальних компонентів, що починається з IK_i, li – індексну послідовність IK_i, що однозначно визначає положення IK_i в ДІК, ld – останню цифру індексної послідовності, $$ – операцію відсікання в індексній послідовності, It – поточну індексну послідовність. Сигнатура піддерева T_{IK_i} розуміється як об'єднання $J(T_{IK_i}) = U (J(BF_{ij}) IK_{ij} \in T_{IK_i})$.</p> <p>Початкові дані: дерево інтелектуальних компонентів T_{IK_0} відповідної ІКСА, що включає множину баз знань $R = \{BF_{ij} i \leq n\}$ де n – число ІК в ДІК; Q – цільова формула з сигнатурою $J(Q)$.</p> <p>Крок 1. Визначається початкова точка процесу логічного виводу. Піддерево T_{IK_i} вибираємо по найближчому ІК для ІО або ІА, що ініціював процес логічного виводу. Встановлюємо li, $It := li$.</p> <p>Крок 2. Проводиться порівняння $J(Q)$ і $J(T_{IK_i})$. Якщо $J(Q) \sqsubset J(T_{IK_i})$, то виведення Q в піддереві T_{IK_i} неможливе. Перехід до кроку 6, інакше до кроку 3.</p> <p>Крок 3. Для всіх $IK_{ij} \sqsubset T_{IK_i}$ виконується порівняння $J(Q)$ і $J(BF_{ij})$. Якщо існує j, такий що $J(Q) \sqsubset J(BF_{ij})$ те застосовується алгоритм MP для множини БЗ піддерева T_{IK_i}. Інакше – до кроку 4.</p>	<p>Обозначим как T_{IK_i} - поддерево дерева интеллектуальных компонентов, начинающееся с IK_i, li - индексную последовательность IK_i, однозначно определяющую положение IK_i в ДИК, ld - последнюю цифру индексной последовательности, $$ - операцию отсечения в индексной последовательности, It - текущую индексную последовательность. Сигнатура поддерева T_{IK_i} понимается как объединение</p> $J(T_{IK_i}) = \bigcup (J(BF_{ij}) IK_{ij} \in T_{IK_i}).$ <p>Исходные данные: дерево интеллектуальных компонентов T_{IK_0} соответствующей КИНС ППР, включающее множество баз знаний $R = \{BF_{ij} i \leq n\}$, где n - число ИК в ДИК; Q - целевая формула с сигнатурой $J(Q)$.</p> <p>Шаг 1. Определяется начальная точка процесса логического вывода. Поддерево T_{IK_i}, выбираем по ближайшему ИК для ИО или ИА, инициировавшего процесс логического вывода. Устанавливаем li, $It := li$.</p> <p>Шаг 2. Производится сравнение $J(Q)$ и $J(T_{IK_i})$. Если $J(Q) \sqsubset J(T_{IK_i})$, то вывод Q в поддереве T_{IK_i} невозможен. Переход к шагу 6, иначе к шагу 3.</p> <p>Шаг 3. Для всех $IK_{ij} \in T_{IK_i}$ выполняется сравнение $J(Q)$ и $J(BF_{ij})$. Если существует j, такой что $J(Q) \sqsubset J(BF_{ij})$, то применяется алгоритм MP для множества БЗ поддерева T_{IK_i}. В противном случае - к шагу 4.</p>
<p>С. 125.</p>	<p>С. 271–272.</p>
<p>Крок 4. Для множини баз знань піддерева T_{IK_i} будується граф перетинів G, далі він перетвориться в дерево GT відповідно до процедури BREAK-CYCLES ($G = (V, E, W)$) [145].</p> <p>Крок 5. Виконується логічний вивід в графі GT. Якщо вдається вивести Q, то видається повідомлення «Мета досягнута», йти до кроку 7, інакше до кроку 6.</p> <p>Крок 6. Перевіряємо рівність $It=0$, тобто чи не досягнутий нульовий рівень ієрархії в ДІК, якщо ні, то $It := It ld$. Перехід до кроку 2. При $It=0$ вивід завершується безрезультатно.</p> <p>Крок 7. Завершення алгоритму MPQ.</p> <p>Для кожної пари $(i, j) \in E$, такої що $i \sqsubset j$ якщо виводиться $BF_j - \phi$ і $J(\phi) \sqsubset J(W(i, j))$, то ϕ додається до BF_i. Відбувається рух процесу логічного виводу від термінальних вершин IK_i в порядку убывання відстані $dist(i, j)$ до вершини дерева IK_i з додаванням в простір пошуку формул ϕ.</p> <p>Адекватність і повноту алгоритму логічного виведення MPQ доводить наступна теорема.</p>	<p>Шаг 4. Для множества баз знаний поддерева T_{IK_i} строится граф пересечений G, далее он преобразуется в дерево GT в соответствии с процедурой BREAKCYCLES ($G = (V, E, W)$) [395].</p> <p>Шаг 5. Выполняется логический вывод в графе GT. Если удается вывести Q, то выдается сообщение «Цель достигнута», идти к шагу 7, иначе к шагу 6.</p> <p>Шаг 6. Проверяем равенство $It = 0$, т.е. не достигнут ли нулевой уровень иерархии в ДИК, если нет, то $It := It ld$. Переход к шагу 2. При $It = 0$ вывод завершается безрезультатно.</p> <p>Шаг 7. Завершение алгоритма MPQ.</p> <p>$\langle \dots \rangle$ Для каждой пары $(j, i) \in E$, такой что $i \sqsubset j$, если выводятся $BF_j - \phi$ и $J(\phi) \sqsubset J(W(i, j))$, то ϕ добавляется к BF_i. Происходит движение процесса логического вывода от терминальных вершин T_{IK_i} в порядке убывания расстояния $dist(i, j)$ к вершине дерева T_{IK_i} с добавлением в пространство поиска формул ϕ.</p> <p>Адекватность и полноту алгоритма логического вывода MPQ доказывает следующая теорема.</p>
<p>С. 125.</p>	<p>С. 272.</p>
<p>Теорема 3.1. Хай T – піддерево інтелектуальних компонентів з сукупністю баз знань $R = \{BF_{ij} i \leq n\}$ для яких побудований граф перетинів G, що є деревом, тобто GT. Хай Q – формула в мові логіки першого</p>	<p>Теорема 4.1. Пусть T - поддерево интеллектуальных компонентов с совокупностью баз знаний $R = \{BF_{ij} i \leq n\}$, для которых построен граф пересечений G, являющийся деревом, т.е. GT. Пусть Q -</p>

<p>порядку, така що $J(Q) \sqsubseteq J(T)$. Тоді $R \models Q$ якщо і тільки якщо алгоритм MPQ виводить "Мета досягнута".</p> <p>Доведення. Для сукупності баз знань $R = \{BF_i\}_{i \leq n}$ дерево $G = (V, E, W)$ називається правильно поміченим, якщо для всіх $(i, j) \in E$ і BF_1, BF_2 з множини R, зв'язаних ребром (i, j) в G, виконується $J(W(i, j)) \sqsubseteq J(BF_1) \cap J(BF_2)$.</p>	<p>формула в языкe логикe первого порядка, такая что $J(Q) \sqsubseteq J(T)$. Тогда $R \models Q$ если и только если алгоритм MPQ выводит «Цель достигнута».</p> <p>Доказательство. Для совокупности баз знаний $R = \{BF_i\}_{i \leq n}$ дерево $G = (V, E, W)$ называется правильно помеченным, если для всех $(i, j) \in E$ и BF_1, BF_2 из множества R, связанных ребром (i, j) в G, выполняется $J(W(i, j)) \sqsubseteq J(BF_1) \cap J(BF_2)$.</p>
<p>С. 126.</p>	<p>С. 272–273.</p>
<p>Скористаємося індукцією по числу баз знань в R. При одній базі знань, коли $R = 1$ алгоритм MPQ співпадає з алгоритмом MP, і отже, теорема справедлива.</p> <p>При двох базах знань, коли $R = 2, R = BF_1 \cup BF_2$, покажемо, що існує формула $\psi \in J(BF_1) \cap J(BF_2)$, така що $BF_1 \models \psi$ і $BF_2 \models \psi \Rightarrow Q$. Використаємо теорему Крейга і позначимо $BF_1 \models \psi = \alpha$, $BF_2 \models \psi \Rightarrow Q = \beta$. По теоремі дедукції для логіки першого порядку $\alpha \vdash \beta$, отже, існує формула $\psi \in J(\alpha) \cap J(\beta)$ така, що $\alpha \vdash \psi$ і $\psi \vdash \beta$. По теоремі дедукції отримуємо, що $BF_1 \models \psi$ і $\psi \& BF_2 \models Q$. Оскільки $\psi \in J(BF_1) \cap J(BF_2)$ по побудові α і β, то теорема справедлива в цьому випадку.</p> <p>Припустимо, що теорема виконується при $R \leq n-1$ і доведемо її справедливості при $R = n$. У графі перетинів n-а вершина може мати s пов'язаних з нею ребер (i_1, i_2, \dots, i_s). Ребро (n, i_1) розділяє дерево G на дві частини: G_1 (включає i_1) і G_2 (включає i_2). Нехай R_1 і R_2 сукупності баз знань, такі що $R_1 \cup R_2 = R$ відповідають дерева G_1 і G_2.</p>	<p>Воспользуемся индукцией по числу баз знаний в R. При одной базе знаний, когда $R =1$, алгоритм MPQ совпадает с алгоритмом MP, и следовательно, теорема справедлива.</p> <p>При двух базах знаний, когда $R =2, R = BF_1 \cup BF_2$, покажем, что существует формула $\psi \in J(BF_1) \cap J(BF_2)$, такая что $BF_1 \models \psi$ и $BF_2 \models \psi \Rightarrow Q$. Используем теорему Крейга и обозначим $BF_1 \models \psi = \alpha$, $BF_2 \models \psi \Rightarrow Q = \beta$. По теореме дедукции для логики первого порядка $\alpha \vdash \beta$, следовательно, существует формула $\psi \in J(\alpha) \cap J(\beta)$, такая что $\alpha \vdash \psi$ и $\psi \vdash \beta$. По теореме дедукции получаем, что $BF_1 \models \psi$ и $\psi \& BF_2 \models Q$. Так как $\psi \in J(BF_1) \cap J(BF_2)$ по построению α и β, то теорема справедлива в этом случае.</p> <p>Предположим, что теорема выполняется при $R \leq n-1$ и докажем ее справедливость при $R =n$. В графе пересечений n-я вершина может иметь s связанных с ней ребер (i_1, i_2, \dots, i_s). Ребро (n, i_1) разделяет дерево G на две части: G_1 (включает i_1) и G_2 (включает n). Пусть R_1 и R_2 совокупности баз знаний, такие что $R_1 \cup R_2 = R$. R_1 и R_2 соответствуют деревья G_1 и G_2.</p>
<p>С. 126–127.</p>	<p>С. 273.</p>
<p>Як було показано вище, якщо $R \models Q$, то існує $\psi \in J(R_1) \cap J(R_2)$ така, що $R_1 \models \psi$ і $R_2 \models \psi \Rightarrow Q$. Оскільки множина ребер (i_1, j) розділяє два підграфу, відповідні теоріям $R_1 \models BF_{i_1}$ і $R_2 \models BF_{i_2}$ виконується</p> $\bigcup_{(i_1, j) \in E, j \neq n} W(i_1, j) \subseteq J(R_2 \cup BF_{i_1}) \cap J(R_1 \models BF_{i_1}).$ <p>Оскільки $\psi \in J(R_1)$ вважаємо, що</p> $\psi \in J(BF_{i_1} \cup \bigcup_{(i_1, j) \in E, j \neq n} W(i_1, j)).$ <p>Але по посилці індукції для G_1, R_1 формула ψ буде доведена в BF_{i_1}, після того, як деякі формули будуть отримані з інших баз в G_1, R_1. Алгоритм MPQ передасть ψ до BF_n, оскільки $\psi \in J(W(i_1, j))$ і G є правильно помічене дерево. Оскільки $R_2 \models \psi \Rightarrow Q$ застосовуючи посилку індукції до $G_2, R_2(\psi \Rightarrow Q \in J(BF_n \cup \bigcup_{(n, i) \in E} W(n, i)))$ отримаємо, що $\psi \Rightarrow Q$ виводиться в BF_n. Таким чином, буде доведено і Q.</p> <p>Проаналізуємо тепер детальніше можливі варіанти виводів в логічній структурі ІККА.</p> <p>Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	<p>Как было показано выше, если $R \models Q$, то существует $\psi \in J(R_1) \cap J(R_2)$, такая, что $R_1 \models \psi$ и $R_2 \models \psi \Rightarrow Q$. Так как множество ребер (i_1, j) разделяет два подграфа, соответствующие теориям $R_1 \models BF_{i_1}$ и $R_2 \models BF_{i_2}$, выполняется</p> $\bigcup_{(i_1, j) \in E, j \neq n} W(i_1, j) \subseteq J(R_2 \cup BF_{i_1}) \cap J(R_1 \models BF_{i_1}).$ <p>Поскольку $\psi \in J(R_1)$, полагаем, что</p> $\psi \in J(BF_{i_1} \cup \bigcup_{(i_1, j) \in E, j \neq n} W(i_1, j)).$ <p>Но посылке индукции для G_1, R_1 формула ψ будет доказана в BF_{i_1}, после того как некоторые формулы будут получены из других баз в G_1, R_1. Алгоритм MPQ передаст ψ к BF_n, т.к. $\psi \in J(W(n, i_1))$ и G есть правильно помеченное дерево. Поскольку $R_2 \models \psi \Rightarrow Q$, применяя посылку индукции к $G_2, R_2(\psi \Rightarrow Q \in J(BF_n \cup \bigcup_{(n, i) \in E} W(n, i)))$ получим, что $\psi \Rightarrow Q$ будет выводима в BF_n. Таким образом, будет доказано и Q.</p> <p>Проанализируем теперь более детально возможные варианты выводов в логической структуре КИНС ППР, в соответствии с направлениями, изложенными в [A41].</p>
<p>С. 127.</p>	<p>С. 274.</p>
<p>Термінальний низхідний неаддитивний вивід</p> <p>Термінальний низхідний неаддитивний вивід (ТННВ) — це логічний вивід, здійснюваний від за-</p>	<p>4.3.1. Термінальний нисходящий неаддитивный вывод</p> <p>Термінальний нисходящий неаддитивний вивід (ТННВ) — это логический вывод, осуществля-</p>

<p>даного ІКі, з послідовним обходом всіх гілок піддерева ІКі без об'єднання БЗ між сусідніми гілками, з подальшими поверненнями у разі недосягнення мети, до тих пір, поки не будуть обійдені всі гілки піддерева ІКі.</p> <p>У разі ТНВВ ми повинні враховувати області дії аксіом і зміст баз фактів, тому схема виводу набуває вигляду</p> $AG, ANG, BF_0 \Rightarrow AG, ANG, AL_1, Con(BF_0, BF_1) \Rightarrow$ $\Rightarrow AG, ANG, AL_1^R, AL_{11}, Con(BF_{11}, Con(BF_0, BF_1)) \Rightarrow \dots$ $\Rightarrow AG, ANG, AL_1^R, AL_{11}^R, \dots, AL_{j1}^R, AL_{j1}^R, Con(BF_{j1}, Con(\dots)\dots)$ <p>де AL_{ij}^R – поширювані вниз локальні аксіоми, Con – операція конкатенації баз фактів. З урахуванням інтерполяційної теореми визначаємо операцію конкатенації таким чином.</p>	<p>емый от заданного ІКі, с последовательным обходом всех ветвей поддерева ІКі без объединения БЗ между соседними ветвями, с последующими возвратами в случае недостижения цели, до тех пор пока не будут обйдены все ветви поддерева ІКі.</p> <p>В случае ТНВВ мы должны учитывать области действия аксиом и содержание баз фактов, поэтому схема вывода приобретает вид</p> $\frac{AG, ANG, BF_0}{\frac{AG, ANG, AL_1, Con(BF_0, BF_1)}{AG, ANG, AL_1^R, AL_{11}, Con(BF_{11}, Con(BF_0, BF_1))}}$ \vdots $\frac{AG, ANG, AL_1^R, AL_{11}^R, \dots, AL_{j1}^R, AL_{j1}^R, Con(BF_{j1}, Con(\dots)\dots)}{\vdots}$ <p>где AL_{ij}^R - распространяемые вниз локальные аксиомы ІКіj, Con - операция конкатенации баз фактов. С учетом интерполяционной теоремы определяем операцию конкатенации следующим образом.</p>
<p>С. 127–128.</p>	<p>С. 274–275.</p>
<p>Нехай BF – поточний стан бази знань процесу логічного виводу, що включає результати попередніх конкатенацій. Якщо процес логічного виводу досяг інтелектуального компонента і з базою даних ВFі то здійснюємо вивід в базі $BF \cup BF_i$ всіх логічних формул, таких що $J(\phi) \sqsubseteq J(W(i, j))$, де j – наступний в ланцюжку виводу (тобто за низхідною ієрархією для ТНВВ) інтелектуальний компонент. Новий зміст BF вважаємо рівним $BF := BF \cup \{\phi\}$, якщо $\{\phi\} = \emptyset$, то поточний стан BF не змінюється:</p> $Con(BF(t), BF_i) = \begin{cases} BF(t+1) = BF \cup \{\phi\}, \{ \phi J(\phi_i) \subseteq J(W(i, j)) \} \\ BF(t+1) = BF(t), \text{ якщо } \{\phi\} = \emptyset \end{cases}$ <p>Оскільки при здійсненні логічного виводу по ДІК необхідне виконання повернень, то вводимо операцію відсікання баз фактів:</p> $Del(BF(t), BF_i) = BF(t-1),$ <p>де t - номер поточного кроку послідовності виводу.</p>	<p>Пусть BF — текущее состояние базы знаний процесса логического вывода, включающее результаты предшествующих конкатенаций. Если процесс логического вывода достиг интеллектуального компонента і с базой данных ВFі, то осуществляем вывод в базе $BF \cup BF_i$ всех логических формул ϕ, таких что $J(\phi) \sqsubseteq J(W(i, j))$, где j — следующий по цепочке вывода (т. е. по нисходящей иерархии для ТНВВ) интеллектуальный компонент. Новое содержание BF полагаем равным $BF := BF \cup \{\phi\}$, если $\{\phi\} = \emptyset$, то текущее состояние BF не изменяется:</p> $Con(BF(t), BF_i) = \begin{cases} BF(t+1) = BF \cup \{\phi\}, \{ \phi J(\phi_i) \subseteq J(W(i, j)) \} \\ BF(t+1) = BF(t), \text{ если } \{\phi\} = \emptyset \end{cases}$ <p>Поскольку при осуществлении логического вывода по ДІК необходимо выполнение возвратов, то вводим операцию отсечения баз фактов:</p> $Del(BF(t), BF_i) = BF(t-1),$ <p>где t - номер текущего шага последовательности вывода.</p>
<p>С. 128.</p>	<p>С. 275.</p>
<p>Недоліком алгоритму МР є вимога того, щоб сигнатура цільової формули належала мові однієї бази знань: $J(Q) \sqsubseteq J(BF_k)$. Це приводить до необхідності порівняння сигнатури $J(Q)$ з сигнатурами всіх BF_j в піддереві виводу з самого початку, що дуже нераціонально. Тому використовується алгоритм МРQ, який припускає можливість пошуку складних цільових формул, коли $J(Q) \sqsubseteq J(BF_k)$.</p> <p>На етапі декомпозиції наочної області виконуємо розподіл тверджень по ІК таким чином: якщо існує формула ϕ, така що $J(\phi)$ використовує імена об'єктів, що належать ІКіj, то ϕ поміщається в ВFі, тобто зв'язується з вищим рівнем ієрархії. Таким чином, формули, що враховують відносини між об'єктами, що знаходяться на різних рівнях ієрархії, зміщуються вгору до кореня дерева декомпозиції.</p> <p>ТНН-вивід здійснюється за наступним алгоритмом.</p> <p>Початкові дані: піддерево ДІК ІКі, мета виводу в попередній нормальній формі (ПНФ) (рис. 3.9).</p>	<p>Недостатком алгоритма МР является требование того, чтобы сигнатура целевой формулы принадлежала языку одной базы знаний: $J(Q) \sqsubseteq J(BF_k)$. Это приводит к необходимости сравнения сигнатуры $J(Q)$ с сигнатурами всех BF_j в поддереве вывода изначально, что весьма нераціонально. Поэтому используется алгоритм МРQ, предполагающий возможность поиска сложных целевых формул, когда $J(Q) \sqsubseteq J(BF_k)$.</p> <p>На этапе декомпозиции предметной области, что детально рассматривается в главе 3, выполняем распределение утверждений по ІК следующим образом: если существует формула ϕ, такая что $J(\phi)$ использует имена объектов, принадлежащих ІКіj, то ϕ помещается в ВFі, т. е. связывается с более высоким уровнем иерархии. Таким образом, формулы, учитывающие отношения между объектами, находящимися на разных уровнях иерархии, смещаются вверх, к корню дерева декомпозиции.</p> <p>ТНН-вывод осуществляется по следующему алгоритму.</p> <p>Исходные данные: поддерево ДІК ІКі, цель вы-</p>

Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.

вода в попередній нормальній формі (ПНФ) (рис. 4.9).

С. 129.

С. 276.

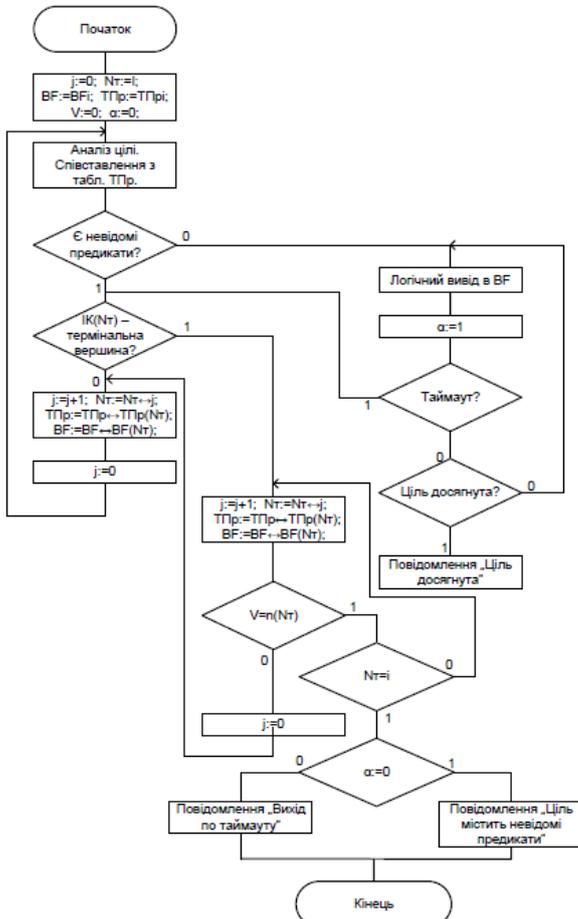


Рис. 3.9. Блок-схема алгоритму ТНН-виводу

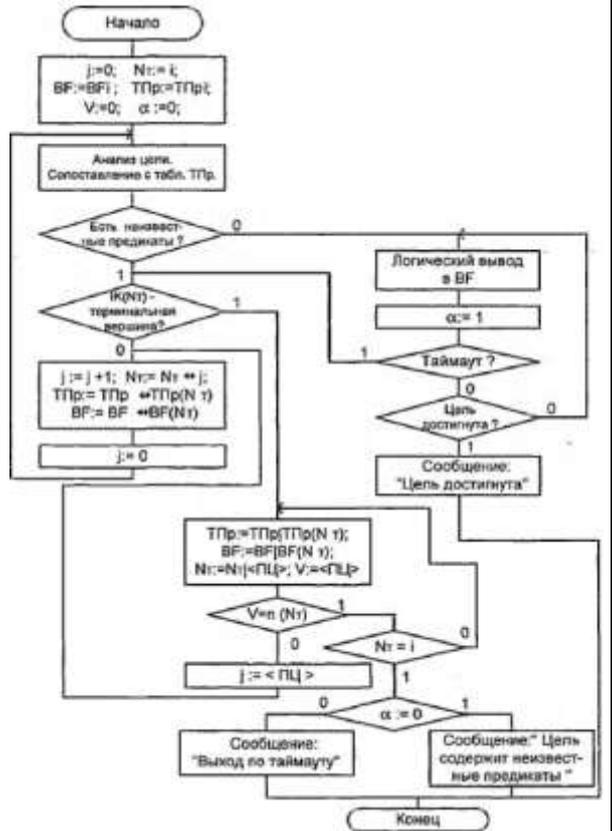


Рис. 4.9. Блок-схема алгоритму ТНН-вывода

С. 130.

С. 275.

Крок 1. Установка початкових параметрів.
Тут NT – номер поточної вершини піддерева IKi, j – приріст номера поточної вершини, BF – сукупність баз фактів, ТПр – сукупність таблиць предикативних символів, V – індекс вершини в піддереві, alpha – ознака здійснення логічного виводу.
Крок 2. Аналіз мети: Якщо мета не містить предикативних символів, що не зустрічаються в поточній сукупності баз фактів, то перехід до кроку 4. Якщо мета містить невідомі предикативні символи, то перевіряється умова: чи є поточна вершина ДІК термінальною? Якщо так, то до кроку 5. Якщо ні, то до кроку 3;
Крок 3. Подовження низхідного ланцюга для подальшого аналізу мети.
Поточний номер вершини нарощується: на одну цифру, значення якої визначається параметром j. Тут ↔ позначає операцію конкатенації. Визначаються база знань і база фактів чергового ІК. Далі приріст j скидається в 0 для подальшого можливого переходу до найлівішої нижчої вершини. Повернення до кроку 2.

Шаг 1. Установка начальных параметров.
Здесь NT - номер текущей вершины поддерева IKi, j - приращение номера текущей вершины, BF - совокупность баз фактов, ТПр - совокупность таблиц предикатных символов, V - индекс вершины в поддереве, alpha - признак осуществления логического вывода.
Шаг 2. Анализ цели. Если цель не содержит предикатных символов, не встречающихся в текущей совокупности баз фактов, то переход к шагу 4. Если цель содержит неизвестные предикатные символы, то проверяется условие: является ли текущая вершина ДИК терминальной? Если да, то к шагу 5. Если нет, то к шагу 3.
Шаг 3. Удлинение нисходящей цепи для последующего анализа цели.
Текущий номер вершины наращивается на одну цифру, значение которой определяется параметром j. Здесь ↔ обозначает операцию конкатенации. Определяются база знаний и база фактов очередного ИК. Далее приращение j сбрасывается в 0 для последующего возможного перехода к самой левой нижестоящей вершине. Возврат к шагу 2.

С. 130–131.	С. 277.
<p>Крок 4. Логічний вивід в сукупності баз фактів ВF. Ознака α встановлюється в одиницю. Вивід закінчується двома способами: або мета досягнута; тоді виводиться результат і перехід на Кінець, або мета не виводиться за встановлений час, тоді відбувається вихід по тайм-ауту.</p> <p>Крок 5. Повернення від термінальної вершини. В цьому випадку пройдено ланцюг вершин від i до деякої термінальної вершини з поточним номером NT і в цього ланцюга залишилися невідомі предикативні символи. Тому потрібно перейти до аналізу найближчої правої вершини. Для цього відскакуються ТПР(NT) і ВF(NT) і остання цифра номера поточної вершини NT, що дозволяє піднятися до найближчої вищої вершини. Тут ПД – остання цифра в параметрі V, вона дозволяє запам'ятати індекс останньої пройденної вершини в піддереві. Якщо значення параметра V дорівнює максимальному індексу (тобто зв'язності по виходу) поточної вершини, то це означає, що піддерево від вершини пройдене і необхідно робити ще один крок вгору, до наступної вершини. При цьому виконується перевірка на досягнення початкової вершини піддерева ІКі. Якщо вона не досягнута, то повторюється крок 5, якщо досягнута, то перехід до кроку 7. Якщо значення параметра V не дорівнює максимальному індексу поточної вершини, це означає, що потрібно переміститися до наступної правої вершини у даному піддереві. Перехід до кроку 6.</p>	<p>Шаг 4. Логический вывод в совокупности баз Фактов ВF. Признак α устанавливается в единицу. Вывод заканчивается двумя способами: либо цель достигнута, тогда выводится результат и переход на Конец, либо цель не выводится за установленное время, тогда происходит выход по тайм-ауту.</p> <p>Шаг 5. Возврат от терминальной вершины. В этом случае пройдена цепь вершин от i до некоторой терминальной вершины с текущим номером NT и в этой цепи остались неизвестные предикативные символы. Поэтому нужно перейти к анализу ближайшей правой вершины. Для этого отсекаются ТПР(NT) и ВF(NT) и последняя цифра номера текущей вершины NT, что позволяет подняться к ближайшей вышестоящей вершине. Здесь ПД – последняя цифра в параметре V, она позволяет запомнить индекс последней пройденной вершины в поддереве. Если значение параметра V равно максимальному индексу (т. е. связности по выходу) текущей вершины, то это означает, что поддерево от вершины NT пройдено и необходимо делать еще один шаг вверх, к следующей вершине. При этом выполняется проверка на достижение исходной вершины поддерева ІКі. Если она не достигнута, то повторяется шаг 5, если достигнута, то переход к шагу 7. Если значение параметра V не равно максимальному индексу текущей вершины, это означает, что нужно переместиться к следующей правой вершине в данном поддереве. Переход к шагу 6.</p>
С. 131.	С. 277.
<p>Крок 6. Установка нового значения прироста номера поточної вершини. Перехід до кроку 3.</p> <p>Крок 7. Аналіз ситуації при поверненні до ІКі. Якщо ознака логічного виводу була встановлена, то це говорить про те, що формула мети коректна, але ні на одній гілці піддерева ІКі цю мету вивести не вдалося при встановленому часі тайм-ауту. Якщо ж ознака логічного виводу не була встановлена, то це означає, що у формулі мети присутні предикати, які або взагалі відсутні в таблиці предикатів піддерева, або знаходяться на різних гілках цього піддерева, і отже, ТНН-вивід не може дати результату.</p> <p>Кінець алгоритму. ТНН-вивід буде глобальним виводом по всьому ДІК у випадку, якщо як початкова вершина вибирається ІК0.</p>	<p>Шаг 6. Установка нового значения приращения номера текущей вершины. Переход к шагу 3.</p> <p>Шаг 7. Анализ ситуации при возврате к ІКі. Если признак логического вывода был установлен, то это говорит о том, что формула цели корректна, но ни на одной ветви поддерева ІКі эту цель вывести не удалось при установленном времени таймаута. Если же признак логического вывода не был установлен, то это означает, что в формуле цели присутствуют предикаты, которые либо вообще отсутствуют в таблице предикатов поддерева ІКі, либо находятся на разных ветвях этого поддерева, и следовательно, ТНН-вывод не может дать результата.</p> <p>Конец алгоритма. ТНН-вывод будет глобальным выводом по всему ДИК в случае, если в качестве начальной вершины выбирается ІК0.</p>
С. 131–132.	С. 278.
<p>Адитивний ієрархічний вивід</p> <p>Адитивний ієрархічний вивід (АІ-вивід) – це логічний вивід, здійснюваний від заданого ІКі з послідовним обходом всіх гілок піддерева ІКі по рівнях низхідної ієрархії з об'єднанням БЗ і таблиць предикативних символів. Алгоритм АІ-виводу складається з двох основних частин. Спочатку виконується аналіз піддерева ІКі з метою розподілу його вершин по рівнях ієрархії, щоб отримати масив такого вигляду (табл. 3.10).</p> <p style="text-align: right;">Таблиця 3.10</p>	<p>4.3.2. Аддитивный иерархический вывод</p> <p>Аддитивный иерархический вывод (АИ-вывод) — это логический вывод, осуществляемый от заданного ІКі с последовательным обходом всех ветвей поддерева ІКі по уровням нисходящей иерархии с объединением БЗ и таблиц предикатных символов. Алгоритм АИ-вывода состоит из двух основных частей. Сначала выполняется анализ поддерева ІКі с целью распределения его вершин по уровням иерархии, чтобы получить массив следующего вида.</p>

	Рівень	Номер вершини		l	Номер вершини
	0	I		0	I
	1	i_1, i_2, \dots, i_n		1	i_1, i_2, \dots, i_n
	2	$i_{11}, i_{12}, \dots, i_{1n}$		2	$i_{11}, i_{12}, \dots, i_{1n}$
			
	G			G	
С. 132.			С. 278–279.		
	<p>Побудувавши такий масив, ми упорядкуємо вершини піддерева ІКі по їх відстані від початкової вершини і визначимо максимальну глибину дерева G. В другій частині алгоритму проводиться послідовне нарощування баз фактів і таблиць предикативних символів по рівнях ієрархії.</p> <p>Алгоритм АІ-виводу представлений на рис. 3.10. Початкові дані: піддерево ДІК ІКі, мета виводу в ПНФ.</p> <p>Крок 1. Установка початкових параметрів.</p> <p>Параметри, що мають однакові позначення з алгоритмом ТНН-виводу, мають той же сенс, тому їх призначення не повторюється. Вводиться двовимірний масив $D(l,m)$ змінної довжини, в якому зберігатиметься структура піддерева ІКі по рівнях ієрархії (тобто по величині відстані в дугах від початкової вершини) ІКі. Ця відстань встановлюватиметься в параметрі l, в масиві $D(l,m)$ – це номер рядка. Параметр m вказуватиме положення елемента в рядку, елемент $D(l,m)$ зберігатиме номер t-ї вершини в l-му рівні ієрархії. Масив $\phi(l)$ зберігатиме поточне значення індекса t для даного рівня l, після проходження всього рівня l елемент $\phi(l)$ міститиме кількість вершин даного рівня ієрархії.</p> <p>Слюняєв переписав чужий текст і чомусь змінив латинську літеру m на українську t. Незнання латинського алфавіту. Недолугий плагіат.</p>			<p>Побудувавши такий масив, ми упорядкуємо вершини піддерева ІКі по їх відстані від початкової вершини і визначимо максимальну глибину дерева G. Во второй части алгоритма проводится последовательное наращивание баз фактов и таблиц предикатных символов по уровням иерархии.</p> <p>Алгоритм АИ-вывода представлен на рис. 4.10. Исходные данные: поддерево ДИК ІКі, цель вывода в ПНФ.</p> <p>Шаг 1. Установка начальных параметров.</p> <p>Параметры, имеющие одинаковые обозначения с алгоритмом ТНН-вывода, имеют тот же смысл, поэтому их назначение не повторяется. Вводится двумерный массив $D(l, m)$ переменной длины, в котором будет храниться структура поддерева ІКі по уровням иерархии (т. с. по величине расстояния в дугах от исходной вершины ІКі). Это расстояние будет устанавливаться в параметре l, в массиве $D(l, m)$ - это номер строки. Параметр m будет указывать положение элемента в строке, элемент $D(l, m)$ будет хранить номер m-й вершины в l-м уровне иерархии. Массив $\phi(l)$ будет хранить текущее значение индекса t для данного уровня l, после прохождения всего уровня l элемент $\phi(l)$ будет содержать количество вершин данного уровня иерархии.</p>	
С. 133.			С. 279.		

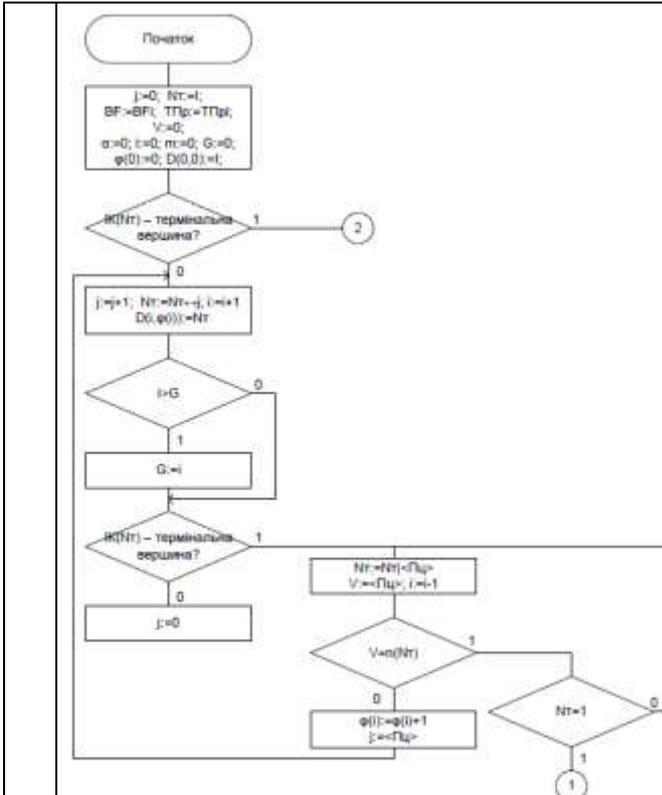


Рис. 3.10. Блок схема алгоритму адитивного ієрархічного виводу (початок)

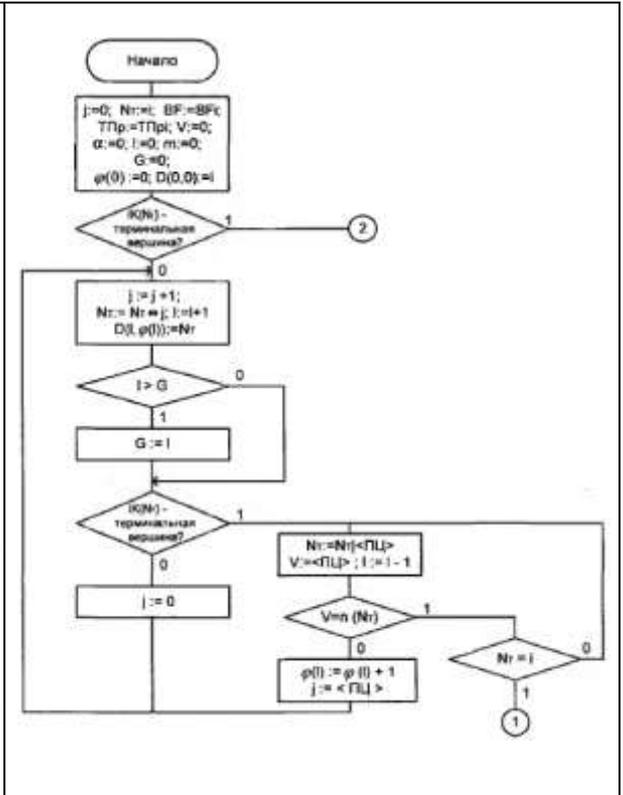


Рис. 4.10. Блок-схема алгоритму АИ-виводу

С. 134.

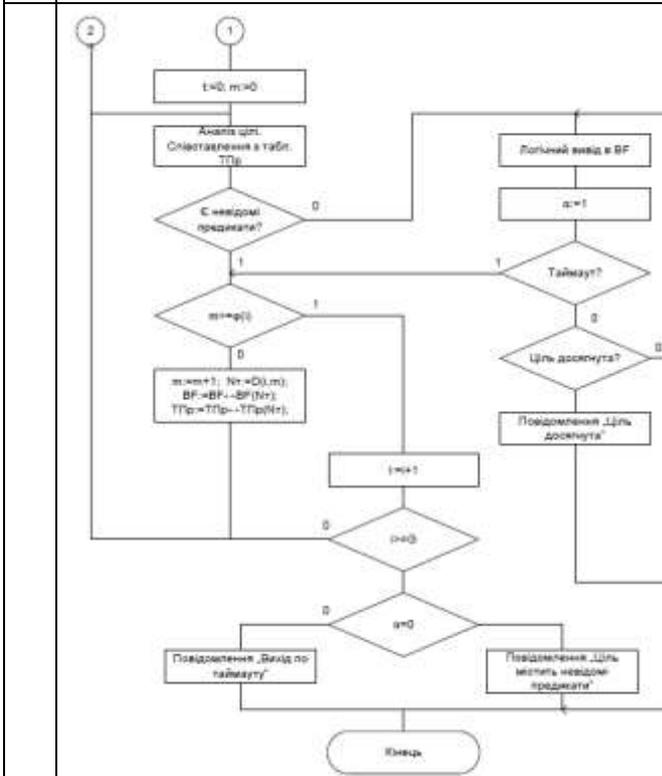


Рис. 3.10. Блок схема алгоритму адитивного ієрархічного виводу (закінчення)

С. 135.

Параметр G служит для збереження максимального рівня, досягнутого для даного дерева ІКІ для того, щоб знати, коли закінчувати обхід рівнів.
Крок 2. Перевірка початкової вершини. В гра-

С. 280.

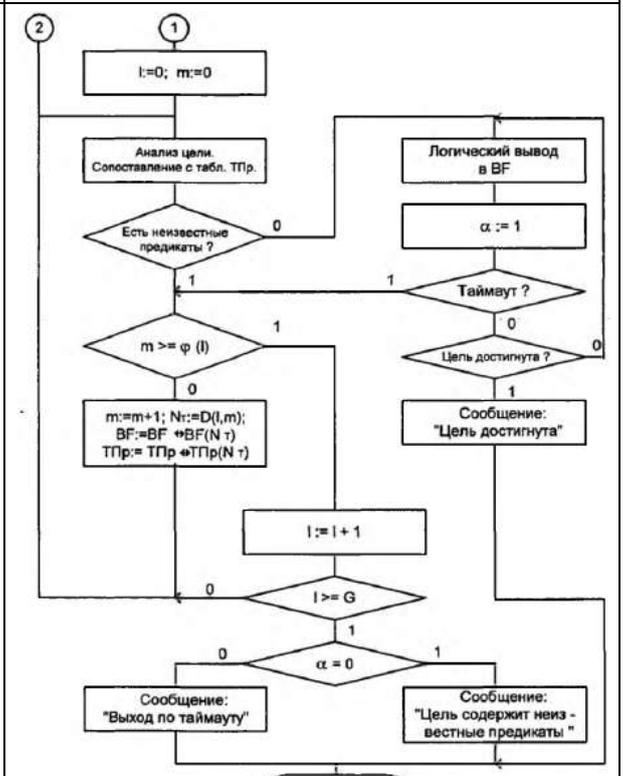


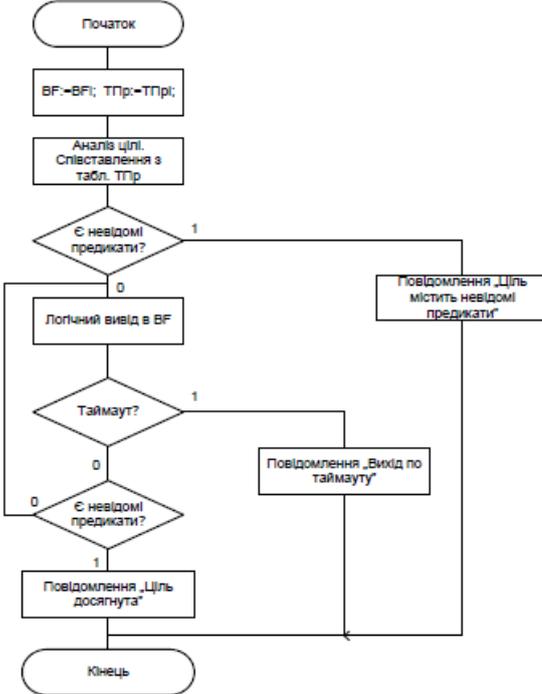
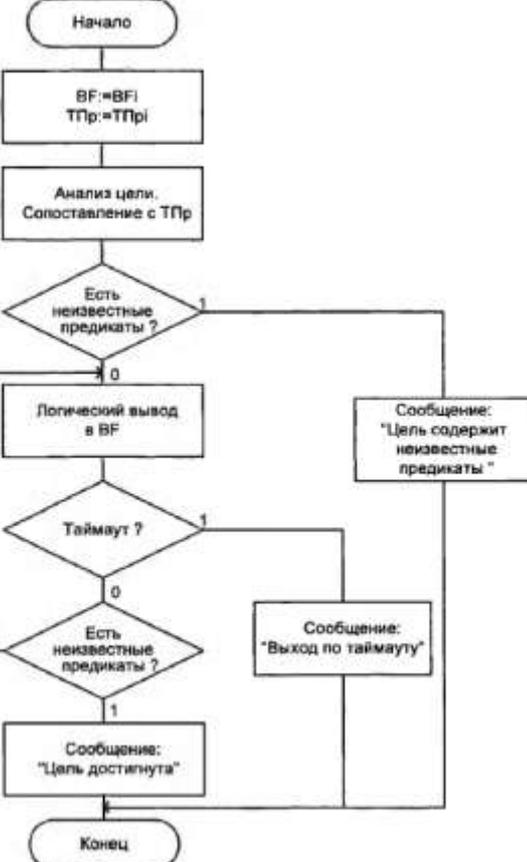
Рис. 4.10. Блок-схема алгоритму АИ-виводу. Продолжение

С. 278, 281.

Параметр G служит для сохранения максимального уровня, достигнутого для данного дерева ІКІ, для того чтобы знать, когда заканчивать обход уровней.

<p>ничному випадку можливо, що ІКі є термінальною вершиною; тому немає сенсу в обході дерева і слід відразу перейти до кроку 7.</p> <p>Крок 3. Подовження низхідного ланцюга. Нарощується поточний номер вершини аналогічно кроку 3 алгоритму ТНН-вывода. Разом з тим нарощується значення відстані l і до масиву $D(l,m)$ записується номер чергової досягнутої вершини відповідно до її рівня ієрархії. Проводиться стеження за відстанню l, якщо досягнуто максимуму, то це значення зберігається в змінній G.</p> <p>Крок 4. Перевірка поточної вершини. Якщо поточна вершина не є термінальною, то слід спуститися на наступний нижчий рівень ієрархії, для чого переміщення скидається в 0 і здійснюється перехід до кроку 3. Якщо поточна вершина є термінальною, то перехід до кроку 5.</p>	<p>Шаг 2. Проверка начальной вершины. В граничном случае возможно, что IK_i является терминальной вершиной, поэтому ног смысла в обходе дерева и следует сразу перейти к шагу 7.</p> <p>Шаг 3. Удлинение нисходящей цепи. Нарращивается текущий номер вершины аналогично шагу 3 алгоритма ТНН-вывода. Вместе с тем наращивается значение расстояния l и в массив $D(l,m)$ записывается номер очередной достигнутой вершины в соответствии с её уровнем иерархии. Проводится слежение за расстоянием l, если достигнут максимум, то это значение сохраняется в переменной G.</p> <p>Шаг 4. Проверка текущей вершины. Если текущая вершина не является терминальной, то следует спуститься на следующий нижестоящий уровень иерархии, для чего перемещение сбрасывается в 0 и осуществляется переход к шагу 3. Если текущая вершина является терминальной, то переход к шагу 5.</p>
<p>С. 135–136.</p>	<p>С. 281.</p>
<p>Крок 5. Повернення від термінальної вершини. Крок в основному аналогічний кроку 5 алгоритму ТНН-выводу, але при цьому декрементується значення відстані l і нарощується індекс рядка $\phi(l)$. Якщо досягнута початкова вершина дерева ІКі, то перехід до кроку 6, якщо ж вона не досягнута, то повторюється крок 5. Якщо значення параметра V не дорівнює максимальному індексу поточної вершини, то повернення до кроку 3.</p> <p>Крок 6. Початок логічного аддитивного виводу. Установка початкових значень масиву вершин $D(l,m)$.</p> <p>Крок 7. Аналіз мети. Якщо мета не містить предикативних символів, які не зустрічаються в поточній сукупності баз фактів, то перехід до кроку 8. Якщо мета містить такі символи, то перевіряється, чи не досягнутий кінець рядка в масиві $D(l,m)$. Якщо він досягнутий, то перехід до кроку 9. Якщо не досягнутий, то переміщуємося до чергової вершини в поточному рядку масиву $D(l,m)$ і нарощуємо сукупності баз значень і таблиць предикативних символів і повертаємося до кроку 7.</p>	<p>Шаг 5. Возврат от терминальной вершины. Шаг в основном аналогичен шагу 5 алгоритма ТНН-вывода, но при этом декрементируется значение расстояния l и наращивается индекс строки $\phi(l)$. Если достигнута исходная вершина дерева IK_i, то переход к шагу 6, если же она не достигнута, то повторяется шаг 5. Если значение параметра V не равно максимальному индексу текущей вершины, то возврат к шагу 3.</p> <p>Шаг 6. Начало логического аддитивного вывода. Установка начальных значений массива вершин $D(l, m)$.</p> <p>Шаг 7. Анализ цели. Если цель не содержит предикатных символов, не встречающихся в текущей совокупности баз фактов, то переход к шагу 8. Если цель содержит такие символы, то проверяется, не достигнут ли конец строки в массиве $D(l, m)$. Если он достигнут, то переход к шагу 9. Если не достигнут, то перемещаемся к очередной вершине в текущей строке массива $D(l, m)$ и наращиваем совокупности баз значений и таблиц предикатных символов и возвращаемся к шагу 7.</p>
<p>С. 136.</p>	<p>С. 281–282.</p>
<p>Крок 8. Логічний вивід в сукупності баз фактів ВФ. Вивід здійснюється до тих пір, поки не буде вичерпано час тайм-ауту або не буде досягнута мета виводу. Якщо спрацьовує ознака тайм-ауту, то виконується повернення в крок 7 для продовження руху по даному рівню ієрархії.</p> <p>Крок 9. Перехід до наступного рівня ієрархії. Інкрементується параметр l і виконується порівняння з максимальним рівнем G. Якщо G не досягнутий, то повернення до кроку 7, інакше – до наступного кроку.</p> <p>Крок 10. Завершення виводу. Аналізується ознака здійснення логічного виводу і формуються відповідні результати.</p> <p>Кінець алгоритму.</p> <p>AI-вивід буде глобальним виводом у разі обрання у якості початкової вершини ІК0.</p>	<p>Шаг 8. Логический вывод в совокупности баз фактов ВФ. Вывод осуществляется до тех пор, пока не будет исчерпано время тайм-аута либо не будет достигнута цель вывода. Если срабатывает признак тайм-аута, то выполняется возврат в шаг 7 для продолжения движения по данному уровню иерархии.</p> <p>Шаг 9. Переход к следующему уровню иерархии. Инкрементируется параметр l и выполняется сравнение с максимальным уровнем G. Если G не достигнут, то возврат к шагу 7, иначе - к следующему шагу.</p> <p>Шаг 10. Завершение вывода. Анализируется признак осуществления логического вывода и формируются соответствующие результаты.</p> <p>Конец алгоритма.</p> <p>AI-вывод будет глобальным выводом в случае взятия в качестве начальной вершины ІК0.</p>
<p>С. 136.</p>	<p>С.282 .</p>
<p>Аддитивний узагальнений вивід</p> <p>Аддитивний узагальнений вивід (АТ-вивід) – це логічний вивід, здійснюваний від заданого з об'єднанням всіх БЗ і всіх таблиць предикативних</p>	<p>4.3.3. Аддитивный обобщенный вывод</p> <p>Аддитивный обобщенный вывод (АО-вывод) — это логический вывод, осуществляемый от заданного ІКі с объединением всех БЗ и всех таблиц</p>

<p>символів в даному піддереві. Початкові дані: піддерево ДІК ІКі, мета виводу в ПНФ. Алгоритм АТ-виводу представлений на рис. 3.11. Використовувані позначення відповідають алгоритмам ТНН-виводу і АІ-виводу, кроки АТ-виводу в основному аналогічні попереднім блок-схемам, тому детальний словесний опис не потрібний. У разі $i = 0$ АТ-вивід здійснюватиметься по всій ІКСА.</p> <p>Локальний вивід</p> <p>Локальний вивід (Л-вивід) — це логічний вивід, здійснюваний в межах даного ІКі. Початкові дані: інтелектуальний компонент, мета виводу в ПНФ. Алгоритм Л-виводу представлений на рис. 3.12.</p>	<p>предикатних символів в даному піддереві. Исходные данные: поддерево ДИК 1К/, цель вывода в ПИФ. Алгоритм АО-вывода представлен на рис. 4.11. Используемые обозначения соответствуют алгоритмам ТНН-вывода и АИ-вывода, шаги ЛО-вывода в основном аналогичны предшествующим блок-схемам, поэтому детальное словесное описание не требуется. В случае $i=0$ АО-вывод будет осуществляться по всей КИНС ППР.</p> <p>4.3.4. Локальный вывод</p> <p>Локальный вывод (Л-вывод) - это логический вывод, осуществляемый в пределах данного ІКі. Исходные данные: интеллектуальный компонент ІКі, цель вывода в ПНФ. Алгоритм Л-вывода представлен на рис. 4.12.</p>
<p>С. 137.</p>	<p>С. 282–283.</p>
<p>Таким чином, запропоновано чотири основні алгоритми логічного виводу в ІКСА, які доведені до необхідного ступеня деталізації і можуть бути реалізовані існуючими інструментальними засобами логічного програмування.</p> <p>Рис. 3.11. Блок-схема алгоритму адитивного узгального виводу</p> <p>Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	<p>Таким образом, предложены четыре основных алгоритма логического вывода в КИНС ППР, которые доведены до необходимой степени детализации и могут быть реализованы существующими инструментальными средствами логического программирования.</p> <p>Рис. 4.11. Блок-схема алгоритма АО-вывода</p> <p>С. .284</p>
<p>С. 138.</p>	<p>С. .284</p>

 <p>Рис. 3.12. Блок-схема алгоритму локального виводу</p>	 <p>Рис. 4.12. Блок-схема алгоритма Л-вывода</p>
<p>С. 138.</p>	<p>С. 285.</p>
<p>Висновки до розділу 3</p> <p>Таким чином запропоновано структурно-логічну модель інтелектуальної інформаційно-керуючої системи аеропорту, що заснована на структурі дерева інтелектуальних компонентів, враховує взаємозв'язки інтелектуальних компонентів, інтелектуальних агентів і інформаційних об'єктів в ієрархічній організації розподіленої інформаційної системи. Логічна модель відображається як в термінах інтелектуальних складових інформаційно-керуючої системи, так і у термінах структур даних, відповідних елементів моделі.</p> <p>Слюняєв пише «Таким чином», нібито це він робить якийсь висновок, а насправді переписує російську дисертацію. Плагіат.</p>	<p>Выводы по главе 4</p> <p>В четвертой главе диссертационной работы предложена структурно-логическая модель КИНС ППР, основанная на структуре дерева интеллектуальных компонентов, учитывающая взаимосвязи интеллектуальных компонентов, интеллектуальных агентов и информационных объектов в иерархической организации распределенной корпоративной системы. Логическая модель отображается как в терминах интеллектуальных составляющих КИНС ППР, так и в терминах структур данных, соответствующих элементам модели.</p>
<p>С. 139.</p>	<p>С. 285.</p>
<p>(Висновки до розділу 3)</p> <p>Для построения архитектуры базы знаний интеллектуально-информационно-керуючої системи обрано змішану стратегію розподілу знань, так як саме ця стратегія забезпечує можливість низхідного стеження і управління станами інтелектуальних компонентів і часткове дублювання аксіоматичних складових баз знань для зниження комунікаційних витрат.</p> <p>На основі даної стратегії розроблено загальну архітектуру бази знань інтелектуально-інформаційно-керуючої системи аеропорту, що включає дерево декомпозиції, моделі інтелектуальних аген-</p>	<p>(Выводы по главе 4)</p> <p>Для построения архитектуры баз знаний КИНС ППР автором выбирается смешанная стратегия распределения знаний, так как именно эта стратегия обеспечивает возможность нисходящего слежения и управления состояниями ИК и частичное дублирование аксиоматических составляющих баз знаний для снижения коммуникационных затрат.</p> <p>На основе данной стратегии разработана общая архитектура баз знаний КИНС ППР, включающая дерево декомпозиции, модели интеллектуальных агентов, модели информационных объектов и базу знаний ФОС. Предложены и обоснованы</p>

<p>тів, моделі інформаційних об'єктів і базу знань формальної об'єктної системи. Запропоновані і обґрунтовані структури даних, необхідні для технічної реалізації розробленої архітектури бази знань.</p> <p>Концептуальною основою розроблених логічних моделей є принцип поділу природного і технічного інтелекту, запропонований В.В. Величенко, А.Н. Швецовим та ін. Виходячи з цього і обґрунтовується вибір дедуктивних методів логічного виводу в архітектурі інтелектуальної інформаційно-керуючої системи аеропорту.</p>	<p>структуры данных, необходимые для технической реализации разработанной архитектуры баз знаний.</p> <p>Концептуальной основой логических моделей, разрабатываемых в четвертой главе диссертационной работы, является принцип разделения естественного и технического интеллекта, предложенный В.В. Величенко. Исходя из этой позиции, в разделе 4.3. обосновывается выбор дедуктивных методов логического вывода в архитектуре КИНС ППР.</p>
<p>С. 139.</p>	<p>С. 285–286.</p>
<p>(Висновки до розділу 3)</p> <p>Удосконалений алгоритм виводу, який базується на відомому алгоритмі МР, відрізняється принципом поділу формул розподіленої бази знань, забезпечує можливість пошуку логічного виведення складних цільових формул, сигнатура яких не належить повністю сигнатурі бази знань одного інтелектуального компонента.</p> <p>З використанням загальнотеоретичного алгоритму дедуктивного виведення розроблено алгоритми: термінального низхідного неадитивного виводу; адитивного ієрархічного виводу; адитивного узагальненого виводу; локального виводу. Дані алгоритми враховують ієрархічну організацію інформаційно-керуючої системи аеропорту та загальну архітектуру бази знань системи, що дозволяє здійснювати виведення рішень на основі інформації різнорідних джерел в умовах динамічної зміни цілей управління.</p>	<p>(Выводы по главе 4)</p> <p>На основе интерполяционной теоремы. Крэйга (W. Craig) предлагается алгоритм вывода, названный автором MPQ, базирующийся на известном алгоритме МР, отличающийся принципом распределения формул распределенной базы знаний, обеспечивающий возможность поиска логического вывода сложных целевых формул, сигнатура которых не принадлежит полностью сигнатуре базы знаний одного ИК.</p> <p>С использованием общетеоретического алгоритма дедуктивного вывода MPQ в разделе 4.3. разработаны следующие алгоритмы:</p> <ul style="list-style-type: none"> - терминального нисходящего неаддитивного вывода; - аддитивного иерархического вывода; - аддитивного обобщенного вывода; - локального вывода. <p>Данные алгоритмы учитывают иерархическую организацию КИНС ППР и архитектуру баз знаний, изложенную в разделе 4.2.</p>