

Методы и средства интеллектуального планирования: применение для управления процессами построения интегрированных экспертных систем¹

Аннотация. Рассматриваются проблемы интеллектуализации процессов разработки интегрированных экспертных систем на основе задачно-ориентированной методологии и инструментального программного комплекса АТ-ТЕХНОЛОГИЯ. Описывается опыт применения методов интеллектуального планирования для синтеза макетов архитектуры прототипов интегрированных экспертных систем на основе использования интеллектуального планировщика, повторно-используемых компонентов, типовых проектных процедур и других компонентов интеллектуальной программной среды комплекса АТ-ТЕХНОЛОГИЯ.

Ключевые слова: искусственный интеллект, интегрированные экспертные системы, задачно-ориентированная методология, модель интеллектуальной среды, инструментальный комплекс АТ-ТЕХНОЛОГИЯ, интеллектуальное планирование.

Введение

Вопросы, связанные с интеллектуальной поддержкой самых трудоемких процессов построения интеллектуальных систем (в частности, приобретение знаний из источников знаний различной типологии, управление проектами и автоматизация процессов проектирования, построение интеллектуальных обучающих систем и др.) всегда занимали важное место в методологиях и технологиях разработки различных классов интеллектуальных систем.

В настоящее время в условиях бурно развивающегося рынка информационных технологий и смещения акцентов в сторону так называемой «открытости» систем как для пользователей, так и для разработчиков, данные проблемы приобретают особую актуальность, поскольку наиболее важными требованиями к методологиям и инструментальным средствам построения интеллектуальных систем становятся факторы снижения стоимостных показателей разработки и интеллектуальной нагрузки на инженеров по знаниям за счет усиления степе-

ни «интеллектуализации» процессов построения систем на всех этапах жизненного цикла. Особенно важное значение эти вопросы имеют для автоматизированной поддержки процессов построения интегрированных экспертных систем (ИЭС) [1], обладающих мощной функциональностью и масштабируемой архитектурой.

Практический опыт, накопленный в ходе создания целого ряда статических, динамических и обучающих ИЭС на основе использования задачно-ориентированной методологии и поддерживающего эту методологию инструментального комплекса АТ-ТЕХНОЛОГИЯ [1-3], показал, что наибольшей сложностью по-прежнему обладают этапы проектирования и реализации ИЭС, причем существенное влияние оказывают специфика конкретной проблемной области и человеческий фактор.

В связи с этим, в контексте дальнейшего развития задачно-ориентированной методологии и комплекса АТ-ТЕХНОЛОГИЯ особенно актуальной является разработка интеллектуальной программной среды и ее базового компонента – интеллектуального планировщика [1]

¹ Работа выполнена при финансовой поддержке РФФИ (проект 15-01-04696).

с целью расширения степени автоматизации процессов планирования и управления проектами, связанными с прототипированием ИЭС. К настоящему времени для комплекса АТ-ТЕХНОЛОГИЯ создано несколько версий интеллектуальных планировщиков, разработанных на основе объединения моделей и методов интеллектуального планирования с методами, традиционно применяемыми в ИЭС [1, 3-7].

В центре внимания данной работы находятся вопросы, связанные с развитием интеллектуального планировщика и других компонентов интеллектуальной программной среды комплекса АТ-ТЕХНОЛОГИЯ, в том числе на основе методов и средств интеллектуального планирования, поэтому важное место уделяется детальному анализу современного состояния исследований и разработок в области интеллектуального планирования (краткий анализ приводился в [8]).

1. Анализ современных методов и программных средств интеллектуального планирования

1.1. Основные термины, подходы, классификации

В настоящее время под *интеллектуальным планированием* чаще всего подразумевается автоматическая генерация планов некоторой программно-аппаратной системой, однако, сам термин «интеллектуальное планирование» не имеет однозначного определения. В русскоязычной литературе по искусственному интеллекту (ИИ) наиболее употребляемым является термин «интеллектуальное планирование» [9], а в современной англоязычной – «автоматическое планирование» (automated planning) [10], причем в обоих случаях имеется в виду собственно процесс генерации плана вычислительной машиной. Соответственно, в контексте интеллектуального планирования *план* - это некоторое представление будущего поведения, в частности, план обычно представляет собой набор действий с некоторыми ограничениями (например, темпоральными) для исполнения некоторым агентом или агентами [10].

Под *планировщиком*, практически, однозначно понимается программная система, осуществляющая генерацию планов на основе формального описания среды, начального состо-

яния среды и поставленных перед планировщиком целей. В некоторых работах планировщик называется агентом [9], в других случаях в состав планировщика входит также *проектировщик*, осуществляющий распределение ресурсов для выполнения построенных планов [10].

В обширной проблематике интеллектуального планирования можно выделить следующие базовые аспекты: *историко-методологический* (выделение и эволюция развития базовых концепций и подходов в области интеллектуального планирования), *формально-концептуальный* (разработка новых моделей, методов, алгоритмов и языков для интеллектуального планирования), *инструментально-технологический* (создание специализированных программных средств для решения задач интеллектуального планирования), *программно-аппаратный* (использование новых технических возможностей для моделирования вычислительно сложных задач интеллектуального планирования), *прикладной* (применение интеллектуального планирования для решения реальных практических задач), *интеграционный* (определяется проблемами интеграции интеллектуальных планировщиков с другими программно-аппаратными системами).

Исходя из вышеперечисленных аспектов, проанализируем особенности наиболее востребованных современных методов и средств интеллектуального планирования (указывая там, где это важно, и хронологию развития).

1.2. Историко-методологический аспект

В контексте историко-методологического аспекта и с учетом фундаментальных работ [9, 10] среди наиболее востребованных и распространенных подходов в настоящее время выделяются [8-11]: планирование с помощью пропозициональной логики; планирование в пространстве планов; планирование в пространстве состояний; планирование как задача удовлетворения ограничений; планирование на основе прецедентов; преобразование в другие задачи/проблемы; темпоральное планирование; планирование в недетерминированных и вероятностных областях; иерархическое планирование (HTN-формализм) и др.

Планирование с помощью пропозициональной логики. Как показано в [11], планирование здесь осуществляется на основе проверки выполнимости логического высказы-

вания, в частности, одним из распространенных примеров этого подхода к решению задачи планирования является широко известная в ИИ система SATPlan, разработанная в конце 1990-х г.г. прошлого века. В последние годы исследования в области эффективных алгоритмов формирования рассуждений для пропозициональной логики привели к возрождению интереса к планированию с помощью логических рассуждений.

Планирование в пространстве планов. Суть планирования на основе поиска в пространстве планов [9] заключается в получении частично-упорядоченных планов для независимого достижения отдельных подцелей, но с учетом пересечений и связей между частичными планами для получения конечного плана. Примером планировщика, реализованного на основе данного подхода, является, в частности, UCSOP [12].

Планирование в пространстве состояний. Классический для ИИ подход, основанный на эвристическом поиске, был успешно реализован в известной системе STRIPS, однако возрождение активного интереса к нему связано с разработкой эффективных эвристических функций, например h^{LM} , h^{FF} , h^{CG} [13], а также *Reachability Heuristic* и *Planning Graph Heuristic* [14], основанных на GraphPlan. В настоящее время поиск в пространстве состояний является наиболее популярным подходом в интеллектуальном планировании, в частности, на его основе реализован планировщик TFD / LAMA [15].

Планирование как задача удовлетворения ограничений. Задача планирования формулируется как задача удовлетворения ограничений (CSP-задача) [9], на основе чего был разработан известный алгоритм GraphPlan, использующий технику прямого распространения ограничений и обеспечивающий перевод описания задачи планирования в специальную структуру - *граф планирования*, из которого впоследствии извлекается частично-упорядоченный план.

Преобразование в другие задачи/проблемы. Здесь осуществляется преобразование исходной задачи планирования в другие задачи, имеющие эффективные решения, например, целочисленное линейное программирование (ILP-задача), CSP-задача, системы линейных уравнений и др. Этот подход реализован в целом ряде планировщиков, в том числе COLIN (COntinuous LINear) и OPTIC [16].

Темпоральное планирование. В отличие от классического планирования, где используется достаточно тривиальная модель времени, в темпоральном планировании [16] за счет использования более сложных моделей времени, учитываются такие параметры, как: продолжительность действий; параллельность планов; ресурсы; динамика изменения среды; привязка целей ко времени и др.

Планирование в недетерминированных и вероятностных областях. Интерес к недетерминированному планированию зародился начиная с 1970-х г.г., однако лишь в последние годы удалось добиться новых успешных результатов. Данный подход позволяет моделировать проблемные области, в которых невозможно однозначно предсказать поведение объекта планирования, поскольку в событиях и действиях используются вероятностные коэффициенты. Реализация здесь может осуществляться как на основе *вероятностных* сетей, так и с помощью *нечетких* сетей, вызывающих наибольший интерес у исследователей. Для данного направления были разработаны языки PPDDL (Probabilistic Planning Domain Definition Language) [17] и пришедшей ему на смену язык Relational Dynamic Influence Diagram Language (RDDL).

Иерархическое планирование (Hierarchical Task Network). Исследования в области HTN-планирования [18] имеют более прикладной характер, в частности, планировщики OPlan, SIP2, SHOP2 [19] и др. сегодня широко используются в различных предметно-ориентированных HTN-системах планирования, а HTN-методы также хорошо комбинируются с другими методами интеллектуального планирования.

Планирование на основе прецедентов (Case Based Planning). Важной особенностью данного подхода (CBP-planning) является использование предшествующего опыта, что обеспечивает следующие преимущества [9]: система планирования становится адаптивной к предметной области (т.е. на основе удачных и неудачных попыток планирования пополняются и/или уточняются знания о предметной области, что позволяет применять подобные системы в динамических средах); использование опыта может повысить эффективность планирования за счет использования при составлении новых планов фрагментов готовых решений (например, после постановки новой задачи

планирования обеспечивается поиск аналогов в библиотеке прецедентов, производится модификация выбранного прецедента путем замены в прецеденте цели и начальных условий, а затем с помощью имитации выполняется проверка полученного решения и выполнение плана в реальной среде). Кроме того, СВР-подход может применяться также в задачах планирования, описанных на языке PDDL [20].

1.3. Формально-концептуальный аспект

Здесь следует отметить, что строгой математической постановки задачи интеллектуального планирования в современных работах, как правило, не приводится, а сама задача рассматривается в контексте решения задачи планирования в сочетании с конкретно выбранными методами ИИ, причем чаще всего, предпочтение отдается использованию методов эвристического поиска в пространстве состояний.

Поэтому рассмотрим наиболее общую постановку задачи планирования [1]. Пусть $I = \{I_k\}$ - множество всех описаний состояний рассматриваемой системы (подмножество окружающего мира). Тогда задачу планирования можно представить как $PlanTask = \langle E, G, A \rangle$, где E является элементом множества I и представляет полное описание начального состояния; G является элементом множества I и представляет описание (возможно, неполное) целевого состояния; $A = \{a_i\}$ - множество действий (операторов, задач), изменяющих состояние системы.

Упорядоченное множество $A = a_1, \dots, a_n$, где для каждого $a_i \in A \rightarrow a_i(E)$, принадлежащего множеству I , называется *решением задачи планирования*, если последовательность операций $a_1(a_2(\dots a_n(I)))$ приводит в итоге к целевому состоянию G . Данное упорядоченное подмножество $\{a_1 \dots a_n\}$ также называется планом.

В современных зарубежных работах (например [10, 18]) для формальной постановки задачи планирования, достаточно часто используется такое базовое понятие, как *система помеченных переходов* (Labeled Transition System), которая описывается в следующем виде [21] $\Sigma = \langle S, A, E, \gamma \rangle$, где $S = \{s_0, s_1, s_2, \dots\}$, - множество состояний; $A = \{a_0, a_1, a_2, \dots\}$ множество действий; $E = \{e_0, e_1, e_2, \dots\}$ множество событий; $\gamma: S \times (A \cup E) \rightarrow 2^S$ - функция перехода. Таким образом, для системы Σ с начальным состоянием S_0 и подмножеством целевых состояний S_g тре-

буется найти последовательность действий, соответствующих переходам между состояниями $(s_0, s_1, \dots, s_k)$ таких, что $s_1 \in \gamma(s_0, a_1)$, $s_2 \in \gamma(s_1, a_2)$, ..., $s_k \in \gamma(s_{k-1}, a_k)$ и $s_k \in S_g$.

В целом, постановки задачи планирования, используемые различными авторами, например [9, 18] и др., опираются на базовый набор следующих аксиом:

- *Конечность Σ* . Система Σ имеет конечное множество состояний.
- *Полная наблюдаемость Σ* . Система Σ полностью наблюдаема, т.е. в ней содержится полная информация о состоянии Σ .
- *Детерминированность Σ* . Система Σ является детерминированной, т.е. для каждого состояния, события или действия существует единственный исход (если действие применимо к состоянию, то его применение переводит детерминированную систему в другое уникальное состояние).
- *Статичность Σ* . Система Σ является статической, т.е. множество событий E пустое;
- *Ограниченность задания целей*. Цель планирования может быть задана только в виде явно заданного состояния системы Σ (одного или нескольких), т.е. исключается возможность задания, например, запрещенного состояния, функции полезности и др.
- *Линейность планов*. Любое решение задачи планирования является линейно упорядоченной конечной последовательностью действий.
- *Неявное задание времени*. Действия и события не имеют длительности и мгновенно переводят систему Σ в новое состояние.
- *Планирование оффлайн*. Планировщик не связан с изменениями, которые могут произойти в Σ в процессе генерации плана.

Следует отметить, что вышеприведенные аксиомы накладывают существенные ограничения на постановку задачи планирования, и, соответственно, нарушение аксиом приводит к усложнению задачи планирования [18].

Таким образом, в области интеллектуального планирования к настоящему времени накоплено значительное число различных моделей, методов и формализмов, в частности, только для моделей планирования [18] может быть построена достаточно объемная онтология, фрагмент которой представлен на Рис. 1. (используется связь единственного типа – наследование).



Рис. 1. Фрагмент онтологии моделей планирования

Прокомментируем кратко приведенный фрагмент онтологии. Использование в качестве одной из вершин онтологии термина *классическое планирование* опирается, в основном, на планирование в системах, в которых выполняются все перечисленные выше аксиомы, причем данный класс задач планирования в литературе чаще всего связывается со STRIPS-планированием, с отсылкой на STRIPS – раннюю систему планирования для описанных выше систем Σ [9, 18]. *Темпоральное* планирование отличается от классического более сложной моделью времени, а именно продолжительностью действий, допустимой параллельностью действий и др. В *ресурсном планировании* кроме булевых переменных появляются вещественные переменные, обеспечивая, таким образом, возможность моделирования более сложных задач. Для *планирования со стоимостью действий* характерна неединичная стоимость действий, хотя данное представление может быть в общем виде промоделировано и в ресурсном планировании. *Планирование с частичным удовлетворением ограничений* характеризуется частичным опи-

санием целей в отличие от классического представления с полностью определенным целевым состоянием. *Недетерминированное* планирование допускает частичную неопределенность как состояний, так и результатов действий, и в настоящее время является наиболее динамично развивающейся моделью планирования.

В целом, алгоритмы планирования основаны на алгоритмах поиска на графах, причем в основе лежит эвристический поиск, а «интеллектуальность» в планировании достигается за счет применения специальных *эвристических функций*, позволяющих эффективно направлять алгоритм эвристического поиска, исключая полный перебор, что дает основания утверждать об *интеллектуальном планировании*. Чаще всего здесь используются общеизвестные алгоритмы, среди которых можно выделить: *A**, *A* Best-First*, *Greedy Best-First*, *Eager Best-First*, *Lazy Best-First*, *Enforced Hill-Climbing*, *Restarting Weighted A**, *And/Or graph (AO*)*, *Loops AO* (LAO*)*, *Bidirectional LAO* (BLAO*)*, *Reversed LAO* (RLAO*)*, *Path Pruning A* (PP-A*)* и др. Базовые версии алгоритмов достаточно

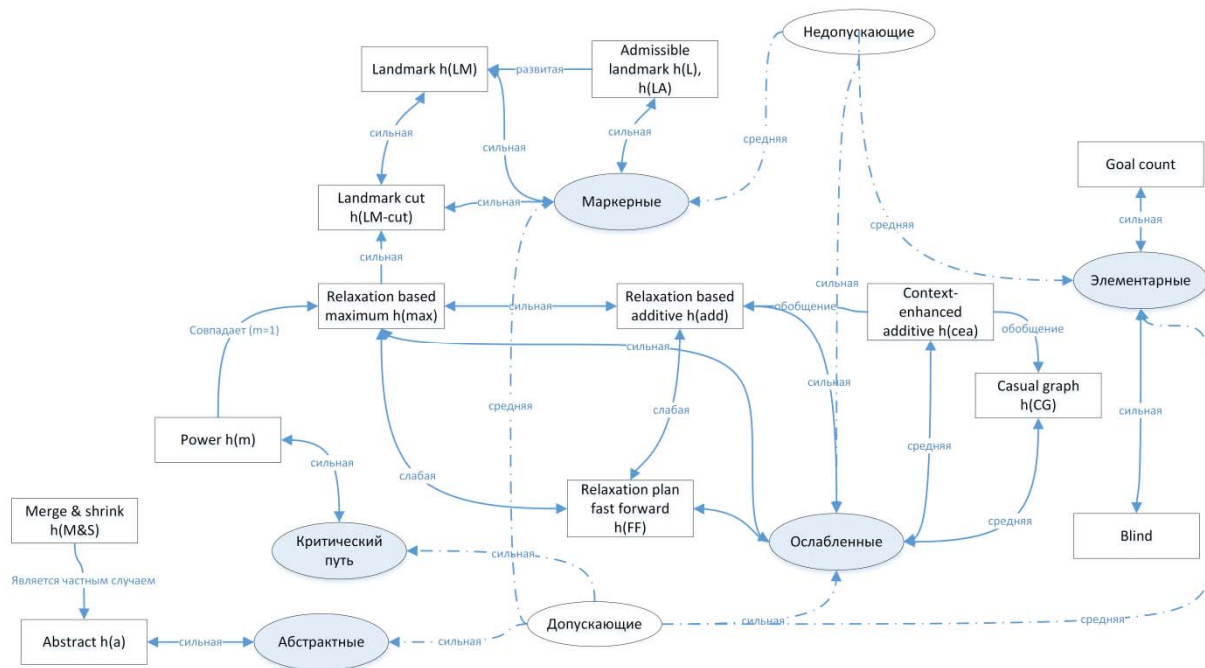


Рис. 2. Фрагмент онтологии эвристических функций, используемых в интеллектуальном планировании

широко описаны, например в [11]. Следует отметить, что в большинстве случаев используются модификации алгоритмов поиска *по первому лучшему совпадению* [11].

В отличие от алгоритмов эвристического поиска, эвристические функции в интеллектуальном планировании имеют достаточно узкую направленность, а их использование обладает одним существенным недостатком - вычислительная сложность эвристического поиска растет экспоненциально в зависимости от размера задачи планирования, поэтому в настоящее время проводятся исследования, связанные с поиском альтернативных путей сокращения пространства состояний, например, за счет применения алгоритма «упрямого набора» [22]. Эвристические функции делятся на *допустимые* (admissible) (A) и *недопустимые* (inadmissible) (NA), и их различие состоит в том, что эвристические функции типа NA могут переоценить реальную стоимость фрагмента оптимального плана, который определяет переход между двумя заданными состояниями.

С учетом [23], можно выделить 4 класса эвристических функций: *ослабленные эвристики*, *эвристики критического пути*, *абстрактные* и *маркерные*. Данная классификация отражена на фрагменте онтологии, представленном на Рис. 2, а ниже приводится Табл. 1, с краткой

характеристикой наиболее часто используемых эвристических функций.

1.4. Инструментально-технологический аспект

В настоящее время своеобразным языковым стандартом в области интеллектуального планирования является разработанный под руководством McDemort язык PDDL (Planning Domain Definition Language) [20], для которого уже созданы три основных версии. В контексте разработки и использования различных формализмов интеллектуального планирования и соответствующих языков, можно выделить следующие тенденции (Рис. 3.): унификация и стандартизация языков планирования на базе языка PDDL; появление проблемно-специализированных языков планирования; смещение акцентов исследований в сторону недетерминированного планирования; появление механизмов для удовлетворения сложных предпочтений и ограничений и др.

Рассмотрим краткую характеристику наиболее востребованных и распространенных языков интеллектуального планирования (Рис.3), причем основное внимание уделим средствам, малоизвестным в отечественной литературе, но широко распространенным в зарубежных разработках.

Табл. 1. Общая характеристика наиболее часто используемых эвристических функций

Название	Тип	Особенности	Авторы и применение
Blind h^0	<i>A</i>	Наиболее простая, не используется на практике	Предложена в [24]
Relaxation-based maximum h^{max}	<i>A</i>	Основывается на поиске плана в <i>ослабленной задаче планирования</i> , т.е. без отрицаний в действиях.	Предложена в [25] в 1999-х гг., использована в системе HAL [24].
h^m	<i>A</i>	Основывается на решении ослабленных задач по достижению подмножества целей исходной задачи, реализация [11] показала низкую производительность.	Предложена в [26] как продолжение развития эвристических функций h^{add} и h^{max} , реализована в HAL [24].
h^a, h^{MKS} (Merge-and-shrink)	<i>A</i>	Основывается на вычислении расстояния между проекциями состояний в абстрактном пространстве.	Предложена в [27], реализована в HAL [24].
Admissible Landmark h^L, h^{LA}	<i>NA</i>	Является развитием h^{LM} [28]. Для получения на ее основе эвристической функции типа <i>A</i> производится определенная модификация.	Предложена в [29] (для формализма SAS+) [30], реализована в LAMA [15] и HAL [24].
h^{LM-cut}	<i>NA</i>	Основывается на <i>маркерах</i> – особых состояниях, которые встречаются в любом плане для заданных целей.	Предложена в [23], как редукция h^{max} , усовершенствована в [31].
Goal count	<i>NA</i>	Вычисляется как количество недостижимых целей.	Реализована в [24]
Relaxation-based additive h^{add}	<i>NA</i>	Является аналогом h^{max} [25], но использует сумму вместо максимума при подсчете стоимости плана для ослабленной задачи.	Использована в системе HAL [24], LAMA [15].
Relaxed plan h^{FF}	<i>NA</i>	Базирована на алгоритме GraphPlan и вычисляется на основе решения ослабленной задачи.	Разработана в [32] для Fast Forward (FF), использовалась в [15, 24, 33].
Casual graph h^{CG}	<i>NA</i>	Основывается на конвертации исходной задачи в формализм SAS+.	Разработана в [13], реализована в HAL [24], использовалась в [33].
Context-enhanced additive h^{ca}	<i>NA</i>	Является обобщением h^{CG} [13] и h^{add} [25].	Предложена в [34], реализована в HAL [24], LAMA [15] и [33].
The Landmark h^{LM}	<i>NA</i>	Является базовой маркерной эвристической функцией, где маркеры извлекаются на основе алгоритма LMRPG (авторы Hoffmann, Porteous and Sebastia).	Предложена в [28], получила развитие в [29], реализована в системах HAL [24], LAMA [15].

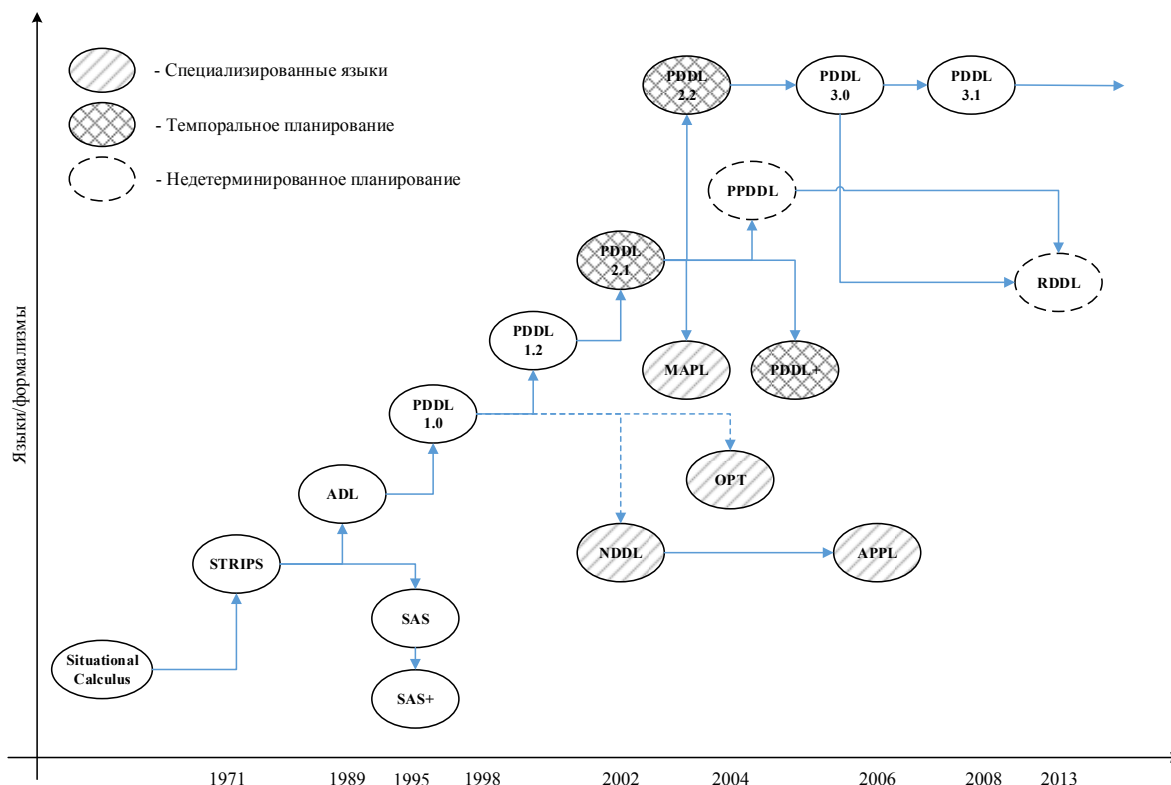


Рис. 3. Хронология развития формализмов/языков интеллектуального планирования

STRIPS. Формализм STRIPS (Stanford Research Institute Problem Solver), разработанный в 1971 г. Р. Файксом и Н. Ниссоном, широко известен в том числе и в отечественной литературе [9]. Несмотря на свои недостатки, STRIPS, являясь базовым расширением PDDL, до сих пор используется для решения некоторых задач, а также в качестве основы для разработки новых алгоритмов планирования [18] и др.

Язык PDDL v1.2 (Planning Domain Definition Language). Язык PDDL, появившийся в 1998 г., достаточно быстро занял лидирующие позиции в качестве универсального языка, позволяющего эмпирически оценивать производительность и эффективность тех или иных методов планирования на соревнованиях планировщиков. PDDL был разработан группой авторов под руководством McDermott D. [20] и впервые представлен на первых международных соревнованиях планировщиков (IPC - International Planning Competition). При помощи PDDL можно описывать *домены планирования* (т.е. модель предметной области без конкретных объектов, значений предикатов и т.д.) и *задачи планирования* (привязанные к определенным доменам). Каждое такое описание оформляется отдельным файлом, т.е. описания домена и задачи физически разделены, что допускает возможность реализации нескольких задач для одного и того же домена.

Домен планирования содержит следующую информацию: типология объектов, используемых в описании задачи планирования; типы отношений между объектами и свойства объектов; константные сущности предметной области; описание схем действий; используемая аксиоматика и др. В свою очередь, в задаче планирования указывается: к какому домену относится данная задача; множество рассматриваемых объектов; начальное состояние (отношения между объектами и их свойства); цель.

Поскольку планировщики обладают различными требованиями к входному языку, то PDDL позволяет описывать требования, используемые в конкретном домене с помощью *флагов-требований*, начинающихся с символа двоеточия (например, возможность типизации объектов, встроенный предикат равенства, условные эффекты и др. флаги).

Язык PDDL v2.1. В следующей версии языка 2.1 появился ряд расширений, направленных на работу с числовыми выражениями (в частности,

для моделирования работы с ресурсами), метриками плана (для поиска оптимального плана) и действиями, имеющими длительность. Другими новациями языка PDDL v2.1. являются возможность работы с такими важными категориями как *время и количество*; более развитые средства для работы с числовыми выражениями (доменные функции); метрики качества плана; действия, имеющие длительность (дискретные и непрерывные); параллелизм действий, имеющих длительность. Отметим, что данный язык был использован в качестве официального языка на третьих международных соревнованиях планировщиков (IPC-2002).

Языки PPDDL и RDDL. Язык PPDDL (Probabilistic PDDL) является расширением PDDL 2.1 и вводит *вероятностные эффекты, вознаграждения, вознаграждающие цели*, и др., что позволяет реализовать планирование на основе MDP-формализма (Markov Decision Process), где есть нечеткость в смене состояний, но среда полностью обозрима для агента/планировщика. PPDDL был официальным языком вероятностной секции на IPC-2004 и IPC-2006. Новый язык RDDL (Relational Dynamic influence Diagram Language), появившийся позднее, концептуально основан на PDDL 1.0 и 3.0, но синтаксически и семантически отличается от обоих, т.к. является более объектно-ориентированным. Базовые идеи языка RDDL близки к формализмам динамических байесовских сетей (DBNs), однако RDDL является более выразительным чем PPDDL. Использовался как официальный язык соревнований IPC-2011 в нечеткой секции.

Схему эволюции систем планирования (планировщиков) можно представить на Рис. 4 и, анализируя общее направление исследований, следует выделить в целом несколько «волн». Первая волна связана с разработкой ранних систем планирования и математического базиса, ставшего основой современных исследований. Вторая волна – это преобладание эвристического поиска и уже достаточно высокий технологический уровень, что позволило развивать методы интеллектуального планирования и подключаться к исследованиям значительному числу научных групп. Характерно, что, несмотря на разнообразие языков и формализмов, исследования проводились автономными коллективами и не существовало общего центра. Третья волна связана с появлением IPC-

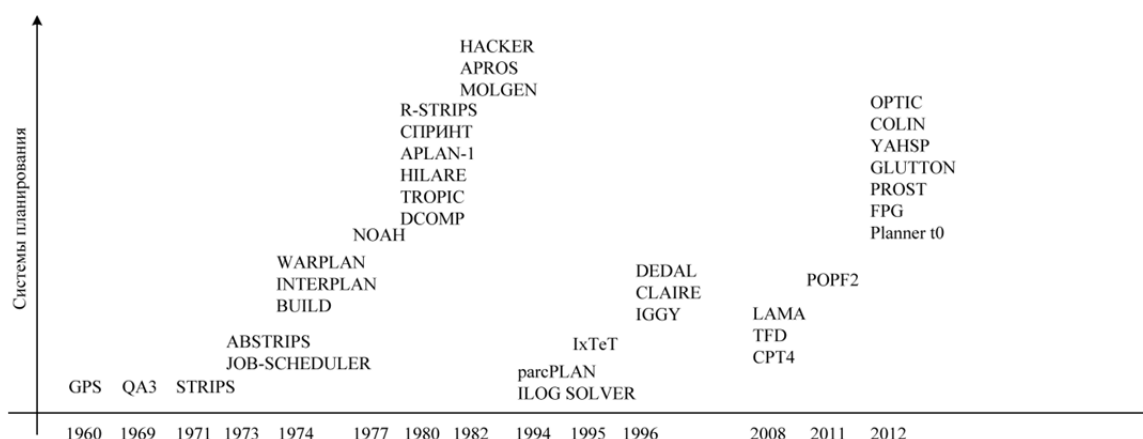


Рис. 4. Хронология развития различных интеллектуальных планировщиков

соревнований в рамках конференций ICAPS, которые создали «место встречи» всех исследователей в данной области, что привело, в частности, к созданию единого языкового стандарта. Основные достижения в контексте данной волны связаны с разработкой эффективных эвристических функций, а также с усилением роли методов недетерминированного планирования.

1.5. Прикладной аспект

С точки зрения областей применения следует отметить, что в настоящее время методы и средства интеллектуального планирования используются в широком классе приложений, наиболее известные из которых кратко рассматриваются ниже [8, 10].

Управление автономными роботами. Управление роботами является одной из традиционных задач интеллектуального планирования. Одним из самых известных и успешных зарубежных проектов является управление марсоходом (проект Mars Rovers [9]), где марсоход управлялся планировщиком на основе данных от различных сенсоров марсохода.

Среди российских исследований следует выделить работы по управлению автономными роботами [36, 37], в частности, расчет оптимальной траектории беспилотных летательных аппаратов, что является достаточно сложной задачей, поскольку необходимо учитывать много аспектов (например, рельеф местности, минимизация расхода топлива, динамика среды, частичная наблюдаемость среды и т.д.). Для решения подобного класса задач применимы методы интеллектуального планирования (в частности, [36, 37], где использовались МТ-графы и HGA* алгоритм для планирования).

Семантический веб. Наиболее актуальной является проблема построения композиции из веб-сервисов для получения новой функциональности. Известны работы, например [38], где интеллектуальное планирование используется для оркестровки веб-сервисов.

Автоматизированное обучение. В образовательной сфере сегодня существует множество традиционных подходов для построения оптимальных индивидуальных планов обучения для обучаемых, однако существенной новацией является использование для этих целей интеллектуального планирования, например работа [39].

Калибровка оборудования. Проблема калибровки большого количества оборудования с минимальными затратами является актуальной на крупных производствах, где используется значительное количество оборудования, требующего периодического обслуживания, и поиск решения данной проблемы существенно зависит от сложности самого процесса калибровки и ограниченности человеческих и временных ресурсов. Имеется успешный опыт применения интеллектуального планирования для подобных задач (например для калибровки картезианского индустриального робота [40]), где планы калибровки позволяли учитывать как нехватку времени, так и ограниченность ресурсов.

Управление конвейерными машинами. На производстве достаточно часто встречаются конвейерные машины, которые управляются минимальным количеством людей (операторов). Как показано в [41] на примере управления конвейером по расфасовке еды в пустую тару, здесь также могут успешно использоваться методы интеллектуального планирования.

Ресурсно-календарное планирование. Ресурсно-календарное планирование является одним из наиболее востребованных подходов в широком спектре прикладных областей - от управления производственным процессом до управления проектной разработкой. Менеджеру проекта, как правило, приходится на основе каких-либо соображений или собственного опыта распределять человеческие ресурсы для решения поставленных задач, и, соответственно, при неправильном распределении повышаются риски того, что проект не будет выполнен в срок. Результаты исследований, приведенные в [42-46], демонстрируют достаточно успешное применение методов интеллектуального планирования для решения задач ресурсно-календарного планирования, в том числе для построения темпоральных планов [16]. Практически, аналогичным образом интеллектуальное планирование применяется для оптимизации ресурсов на производстве, например [47].

Инженерия знаний. Здесь интеллектуальное планирование применяется в основном в области DataMining, например, работа [48], в которой описан опыт построения планов извлечения знаний и моделирования самих процессов извлечения. Также интеллектуальное планирование применяется для управления процессом DataMining, например [49].

Распределение ресурсов в вычислительных системах. Проблема распределения ресурсов особенно актуальна в системах реального времени, где необходимо гарантировать заданное время отклика, причем сложность проблемы возрастает за счет дополнительных накладных расходов, связанных с особенностями программного и/или аппаратного распределения ресурсов конкретных систем. Один из подходов к решению данной проблемы с помощью применения интеллектуального планирования демонстрируется в работе [50].

Логистика. В современных технологиях управления логистической инфраструктурой, а также при моделировании логистических сетей, пассажирских перевозок, цепей поставок и др., прослеживается усиление тенденций к использованию методов интеллектуального планирования, о чем, в частности, свидетельствует опыт международных ИРС-соревнований планировщиков (например, в качестве одной из групп тестовых заданий для планировщиков являлась совокупность логистических задач [51]).

Автоматизация разработки программного обеспечения. Здесь следует выделить отдельные работы [52], связанные с попытками применения методов интеллектуального планирования для синтеза программного обеспечения. Имеется опыт решения проблемы синтеза линейных, ветвящихся и рекурсивных структур управления с точки зрения эффективной реализации алгоритмов синтеза линейных функциональных программ, рекурсивных программ, программ с подпрограммами и параллельных программ.

Завершая краткий анализ основных областей применения методов и средств интеллектуального планирования, следует отметить, что в целом их использование с целью автоматизированной поддержки процессов построения интеллектуальных систем является пока слабо исследованной областью. Здесь можно сослаться, в основном, на опыт, накопленный в ходе создания прикладных ИЭС на основе задачно-ориентированной методологии и поддерживающего ее инструментария – комплекса АТ-ТЕХНОЛОГИЯ [1-8, 35]. Рассмотрим более детально основные концепции интеллектуальной программной среды комплекса АТ-ТЕХНОЛОГИЯ.

2. Особенности построения интегрированных экспертных систем на основе задачно-ориентированной методологии

2.1. Модель интеллектуальной программной среды и ее компоненты

Значительное место в рамках задачно-ориентированной методологии отводится методам и средствам интеллектуальной программной поддержки процессов разработки, которые объединяются общим понятием «интеллектуальная среда» (основные положения методологии отражены в монографии [1] и других работах, например [4, 6]). Полное формальное описание модели интеллектуальной среды и методов реализации отдельных компонентов представлено в [1], поэтому здесь приведем только краткое описание этой модели в виде четверки $M_{AT} = \langle KB, K, P, TI \rangle$, где KB – технологическая база знаний (БЗ) о составе проекта и типовых проектных решениях, используемых

при разработке ИЭС; $K=\{K_i\}$, $i=1..m$ – множество текущих контекстов K_i , состоящих из множества объектов из KB , редактируемых или выполняющихся на текущем шаге управления; P – специальная программа – интеллектуальный планировщик, управляющая процессами разработки и тестирования ИЭС; $PI=\{PI_i\}$, $i=1..n$ – множество инструментов PI_i , применяющихся на различных этапах разработки ИЭС.

Декларативной основой процесса интеллектуальной поддержки разработки ИЭС является компонент KB , выступающий в качестве информационного хранилища в данной среде и определяемый как $KB=\langle WKB, SKB, PKB \rangle$, где WKB – это БЗ, содержащая знания о типовых проектных процедурах (ТПП), описывающих последовательности и способы применения тех или иных инструментальных средств при создании прикладных ИЭС, а также последовательности этапов создания ИЭС; SKB – это БЗ, включающая знания об использовании ТПП и повторно-используемых компонентов (ПИК), включая фрагменты созданных ранее прототипов ИЭС; PKB (опционально) – представляет собой БЗ, содержащую специфические знания, используемые на различных этапах создания прототипа ИЭС для решения задач, требующих нестандартного подхода.

Текущий контекст K_i представляется в виде совокупности $K_i=\langle KD, KP \rangle$, где KD – декларативный контекст, предназначенный для хранения статической декларативной информации о структуре проекта, инженеру по знаниям и текущем пользователе; KP – процедурный контекст, включающий в себя объекты, явно влияющие на дальнейшие шаги планировщика (этап ЖЦ системы, текущий редактируемый или исполняемый объект, текущая цель, текущий исполнитель, глобальный план разработки и т.д.).

Основной процедурный (операционный) компонент – интеллектуальный планировщик в общем виде описывается моделью $P=\langle SK, AF, Pa, Pb, I, GP \rangle$, где SK – состояние текущего контекста, при котором активизируется планировщик; $AF=\{AF_i\}$, $i=1..k$ – множество функциональных модулей AF_i , входящих в состав планировщика; Pa – процедура выбора текущей цели на основании глобального плана разработки; Pb – процедура выбора наилучшего функционального модуля-исполнителя из списка возможных кандидатов; I – процедуры, обеспечивающие интерфейс с соответствующими

компонентами комплекса АТ-ТЕХНОЛОГИЯ; GP – процедуры работы с глобальным планом разработки ИЭС.

Модель любой ТПП представляется в виде тройки: $TPP_i=\langle C, L, T \rangle$, где C – множество условий, при выполнении которых возможна реализация ТПП; L – сценарий выполнения, описанный на внутреннем языке описания действий ТПП; T – множество параметров, инициализируемых интеллектуальным планировщиком при включении ТПП в план разработки прототипа ИЭС. Каждый ПИК, участвующий в разработке прототипа ИЭС, представляется пятеркой вида $PIK=\langle N, Arg, F, PINT, FN \rangle$, где N – имя компонента, под которым он зарегистрирован в комплексе; $Arg=\{Arg_i\}$, $i=1..l$ – множество аргументов, содержащих поддеревья базы данных текущего проекта, служащие входными параметрами для выполнения функций из множества $F=\{F_i\}$, $i=1..s$ – множество методов (интерфейсов ПИК) данного компонента на уровне реализации; $PINT$ – множество наименований интерфейсов других ПИК, используемых методами данного ПИК, $FN=\{FN_i\}$, $i=1..v$ – множество наименований функций, выполняемых данным ПИК.

Модель процесса функционирования прототипа ИЭС представлена в виде тройки: $M_A=\langle Sc, C, Cl \rangle$, где Sc – сценарий работы прототипа ИЭС; C – множество подсистем прототипа ИЭС, которые можно условно поделить на 2 категории (стандартные подсистемы (ПИК) из репозитория ПИК и подсистемы, реализованные разработчиками прототипа ИЭС); Cl – отношение «передача управления», описывающее порядок взаимодействия между подсистемами прототипа ИЭС.

Рассмотрим кратко методы и подходы, использованные при реализации модели интеллектуальной среды поддержки разработки ИЭС, основными компонентами которой являются технологическая БЗ о составе проекта ИЭС, ТПП и ПИК, а также интеллектуальный планировщик, управляющий процессами построения и исполнения планов разработки прототипов ИЭС. Перечислим основные цели, для достижения которых необходимо использовать знания различных типов в процессе разработки прототипа ИЭС: проверка ссылочной целостности проекта по разработке ИЭС; автоматизированное построение диаграммы компонентов; синтез макета архитектуры прототипа ИЭС;

планирование последовательности шагов по созданию прототипа ИЭС с учетом специфики и особенностей решаемых задач; определение множества наиболее актуальных подзадач для каждого из этапов (шагов) в ходе разработки прототипа ИЭС и др.

2.2. Построение плана разработки прототипа прикладной ИЭС

Задача построения плана разработки прототипа прикладной ИЭС представляет собой полноценную задачу из области ИИ, поскольку требует привлечения самых разных знаний о моделях и методах решения типовых задач [1], о технологии проектирования и разработки ИЭС, о способах интеграции с внешними базами данных и пакетами прикладных программ и др. Поэтому проектом по разработке ИЭС на основе задачно-ориентированной методологии называется хранящаяся в некотором формате на физическом носителе совокупность знаний и данных о решаемой задаче, на основе которых под управлением интеллектуального планировщика осуществляется процесс прототипирования прикладной ИЭС.

Следует отметить, что реализация текущей версии интеллектуального планировщика [35] представляет собой гибридизацию подходов, базирующихся на использовании HTN-формализма и гибких механизмов поиска решений, применяемых в ИЭС, что позволяет использовать декларативный способ описания знаний о ходе разработки, в данном случае на языке представления знаний продукционного типа [1].

Для описания базовых процессов управления разработкой прототипом ИЭС, реализация которых требует привлечения отдельных видов знаний и использования интеллектуального планировщика, приведем в соответствии с [1] краткое описание некоторых основных моделей.

Модель прототипа ИЭС представляется как семерка вида: $PRJ = \langle PN, KB, Solver, Pd, PDFD, PPIK, PCOMP \rangle$, где PN – имя проекта, KB – БЗ прототипа ИЭС, $Solver$ – машина (средства) вывода прототипа ИЭС; Pd – данные проекта, т.е. информация различного характера (знания, данные, отдельные параметры, тексты и т.д.), используемая интеллектуальным планировщиком как в процессе разработки прототипа ИЭС, так и для генерации готового прототипа, причем к основным данным проекта можно отнести протоколы интервьюирования экспертов,

словарь лексики, фрагменты поля знаний, БЗ в разных языках представления знаний, тип решаемой задачи, а также различную служебную информацию (профиль текущего пользователя, имя инженера по знаниям, создавшего проект, даты начала и предполагаемого завершения и т.д.); $PDFD$ – расширенная информационно-логическая модель архитектуры прототипа ИЭС в виде иерархии расширенных диаграмм потоков данных (РДПД), являющаяся одной из важнейших составляющих проекта, поскольку ее структура во многом определяет состав прототипа и его функциональные возможности (иерархия РДПД строится с использованием всего одного отношения – отношения декомпозиции, т.е. операция верхнего уровня детализируется с помощью РДПД нижнего уровня иерархии, причем возможно существование только одной РДПД, не являющейся детализацией какой-либо операции, например, «контекстная диаграмма потоков данных»; – совокупность ПИК; $PCOMP = \{PCOMP_i\}$, $i = 1..I$ – совокупность различных подсистем ИЭС, разработанных как средствами комплекса АТ-ТЕХНОЛОГИЯ, так и с помощью внешних приложений.

Модель плана разработки прототипа ИЭС представлена пятеркой вида: $PL = \langle S, PP, A, R, P, D \rangle$, где S – множество стадий разработки прототипа ИЭС, под которыми обычно понимаются этапы жизненного цикла (ЖЦ) создания прототипа ИЭС; PP – множество ТПП; A – множество задач, реализация которых инженером по знаниям необходима для успешной разработки прототипа ИЭС; R – отношение «входить в состав» между элементами множества A_i из множества A ; P – отношение следования между задачами A_i из множества A ; D – отношение, определяющее выполнимость/невыполнимость задачи A_i из множества A с помощью конкретной ТПП из множества PP . Для плана PL с помощью HTN-формализма может быть определена иерархическая сеть задач в виде $TN = \langle A, R \rangle$.

План разработки прототипа ИЭС, в свою очередь, может быть декомпозирован на глобальный и детальный. Глобальный план PLG представляет собой $PLG = \{AG, PG, CA\}$, где $AG \subseteq A$; $PG \subseteq P$, определенное на множестве AG ; CA – отношение «соответствовать компоненту модели архитектуры», связывающее элементы из множества AG с компонентами модели архитектуры прототипа ИЭС из $PDFD$, описанной в виде иерархии

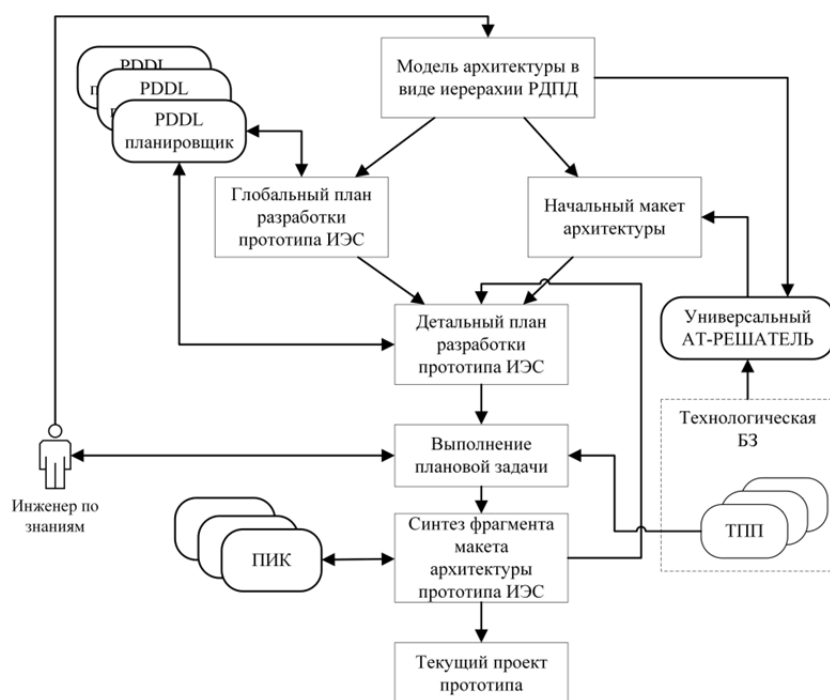


Рис. 5. Процесс построения прототипов ИЭС на основе интеграции методов планирования с методами ИЭС

РДПД. Следует отметить, что разработка каждого компонента из модели архитектура прототипа ИЭС представляет собой сложный процесс, который может быть детализирован полноценной сетью конкретных задач. Детальный план *PLD* представляет собой детализацию глобального плана *PLG* на основе HTN-формализма, и определяется как $PLG = \{A, P, TN\}$.

Под макетом архитектуры прототипа ИЭС понимается множество взаимосвязанных ПИК и иных подсистем, решающих задачу, описанную с помощью информационно-логической модели на этапе анализа системных требования пользователя.

Приведем общее описание задачи синтеза плана разработки прототипа ИЭС. Пусть K_U – множество всех возможных контекстов K . Тогда задачу планирования разработки прототипа ИЭС можно представить как $PlanningTask = \langle K, A_U, K_G \rangle$, где A_U – множество всех доступных действий инженера по знаниям над проектом PRJ , K – является элементом K_U и имеет полное описание, K_G – является элементом K_U и представляет описание (возможно, не полное) целевого контекста проекта. Упорядоченное множество $DevPlan = ap_1 \dots ap_n$, где каждый $ap_i \in A_U$, называется решением задачи планирования, если суперпозиция операций $ap_n(\dots(ap_2(ap_1(K)))\dots)$ приводит в итоге к целе-

вому состоянию K_G . Описанная ранее модель плана *PL* является развитием данной модели *DevPlan* на основе HTN-формализма и с учетом дополнительной информации.

Таким образом, как показано на Рис. 5, основной задачей интеллектуального планировщика является динамическая поддержка действий инженера по знаниям на всех этапах ЖЦ построения прототипов ИЭС с помощью генерации планов разработки текущих прототипов ИЭС и обеспечение возможностей исполнения конкретных планов (производится как в автоматическом режиме, так и в интерактивном режиме). Следует отметить, что генерация детального и глобального планов построения прототипов ИЭС, а также синтез макета архитектуры осуществляется на основе интеграции методов ИЭС с методами планирования.

Поскольку задача планирования разработки прототипа ИЭС, как правило, подразумевает однозначный исход выполнения плановых задач, то в качестве основного подхода было выбрано *детерминированное планирование*. Конечно, подобный подход не позволяет в явном виде учесть события, связанные с рисками и срывами проекта вследствие человеческого фактора, но это компенсируется стратегией перестройки плана в случае возникновения отклонений от плана. При этом среда для планировщика является полно-

стью обозримой и представляет собой описание проекта прототипа ИЭС.

Планирование выполняется в *пространстве состояний* проекта (в ранних работах использовалось понятие «пространство проектирования»), которое формируется за счет множества возможных значений параметров проектов. Выбор данного подхода обусловлен популярностью и наличием большого количества инструментальных средств, которые можно эффективно интегрировать в интеллектуальный планировщик комплекса АТ-ТЕХНОЛОГИЯ.

В целом план построения прототипа ИЭС представляет собой полноценную задачу проектной разработки, поэтому элементы плана необходимо привязывать ко времени, в соответствии с чем был выбран *темпоральный подход* с явным моделированием времени, поскольку в дальнейших исследованиях планируется адаптировать интеллектуальный планировщик под управление командной разработкой, что является еще одним аргументом в пользу темпорального планирования. Следует добавить, что в качестве основы для объединения данных подходов используется описанный выше формализм и язык PDDL 2.1, позволяющий учитывать время и использовать вещественные переменные.

3. Особенности реализации компонентов модели интеллектуальной программной среды

Инструментальный комплекс АТ-ТЕХНОЛОГИЯ третьего поколения [1, 3, 4] представляет собой достаточно сложное современное программное средство, предназначенное для решения задач, связанных с проектированием, разработкой и сопровождением статических и динамических ИЭС. В основу реализации текущей версии комплекса положен подход, опирающийся на технологию использования динамических расширений (плагинов), обеспечивающих модульность комплекса и легкую взаимозаменяемость дополнительных инструментов разработчика (инженера по знаниям), реализуемых в виде ПИК комплекса. Ядро комплекса включает совокупность базовых компонентов, необходимых для создания ши-

рокого класса прикладных ИЭС на основе задачно-ориентированной методологии.

Интеллектуальная среда поддержки разработки ИЭС реализована в рамках архитектуры комплекса АТ-ТЕХНОЛОГИЯ в виде нескольких программных модулей (интеллектуальный планировщик, менеджер проекта и т.п.) и информационных хранилищ (репозиторий проекта, технологическая БЗ и т.п.). Интеллектуальный планировщик содержит компонент взаимодействия с инженером по знаниям, предназначенный для отображения плана разработки и доступных действий, которые могут быть выполнены на текущем этапе (шаге) разработки и при определенных условиях, сложившихся в проекте. Кроме этого планировщик отображает весь план разработки прототипа ИЭС и реализует обратную связь с инженером по знаниям путем обработки активированных интерактивных задач, осуществляя выполнение конкретных действий посредством связи с ПИК комплекса.

3.1. Особенности реализации интеллектуального планировщика

Программная реализация текущей версии интеллектуального планировщика выполнена в виде нескольких блоков, основными из которых являются – компонент синтеза макета архитектуры, визуализатор плана, компонент взаимодействия с инженером по знаниям, компонент генерации планов разработки ИЭС и др. (Рис.6). Рассмотрим особенности реализации наиболее важных компонентов.

Визуализатор плана. Предназначен для инженера по знаниям с целью визуализации текущих планов и инициирования выполнения определенной задачи плана. Визуализация осуществляется в текстовом и графическом виде, причем с помощью графической визуализации для максимальной наглядности план отображается в виде двухмерной графической сцены (с использованием библиотеки QT).

Компонент генерации плана разработки прототипа ИЭС. Обеспечиваются процессы генерации обобщенных планов (на основе модели архитектуры ИЭС) и детальных планов (с использованием текущего макета архитектуры ИЭС и обобщенного плана). Реализация данных процессов осуществляется с помощью

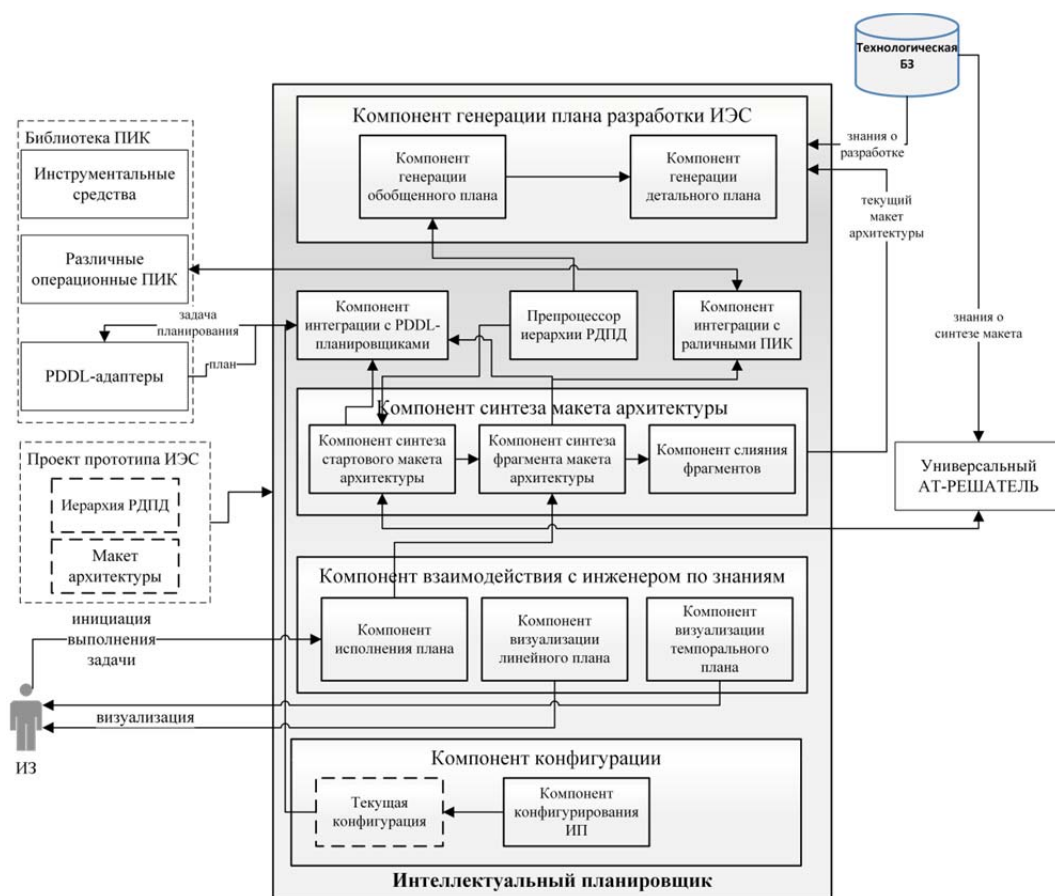


Рис. 6. Архитектура интеллектуального планировщика комплекса AT-ТЕХНОЛОГИЯ

взаимодействия с внешними PDDL-планировщиками через компонент интеграции.

Компонент интеграции с PDDL-планировщиками. Для решения задачи планирования интеллектуальный планировщик использует внешние PDDL-планировщики, в том числе LAMA, COLIN [12, 37]. Процесс взаимодействия удобно рассмотреть по слоям. Компонент интеграции используется как при генерации глобального, так и при генерации детального планов, причем в обоих случаях для этого используются внешние PDDL-планировщики.

Для решения с помощью интеллектуального планировщика описанной выше задачи планирования, используются методы интеллектуального планирования - *детерминированное классическое планирование в пространстве состояний с тривиальной моделью времени* (для генерации глобального плана), а также *темпоральное планирование со стоимостями действий* (для генерации детального плана). Данные методы реализуются с помощью внешних программ – PDDL-планировщиков, инте-

грация с которыми производится на основе файлового обмена. Для PDDL-планировщиков формируется задача на языке PDDL версии 2.1 с дополнительными расширениями, которая далее решается в виде отдельного процесса.

Разделение на глобальный и детальный план позволяет эффективно оптимизировать функционирование интеллектуального планировщика, сохранив его время отклика, так как количество состояний даже для небольшой задачи планирования может быть достаточно велико [11]. Поскольку задачу планирования для PDDL-планировщика необходимо представить в виде четверки $\Pi = \langle A_\Pi, I_\Pi, O_\Pi, G_\Pi \rangle$ [36], где A_Π – множество переменных, определяющих состояние; I_Π – начальное состояние; O_Π – множество операторов; G_Π – целевое состояние, то с помощью интеллектуального планировщика производится преобразование задачи планирования *PlanningTask* (раздел 2.2) следующим образом: из PRJ , K_U и отношения R (для определения иерархической связи между задачами) формируется набор PDDL-предикатов A_Π ; из

PRJ и текущего контекста K формируется множество истинных предикатов и значение метрики I_L ; из A_U и PRJ формируется множество PDDL-операторов (действий) O_L ; из целевого контекста K_G и PRJ формируется множество истинных для целевого состояния PDDL-предикатов. Следует отметить, что множество операторов O_L задается на языке PDDL неявно за счет использования типизации и дополнительных предикатов.

Компонент синтеза макета архитектуры.

Производится синтез начального макета архитектуры на основе модели архитектуры прототипа ИЭС (представленной в виде иерархии РДПД), синтез фрагментов макета архитектуры путем выполнения задач плана, а также слияние нового фрагмента архитектуры с текущим макетом. Синтез начального макета производится с учетом специфики проблемной области, для которой разрабатывается прототип ИЭС, в связи с чем для построения начального макета архитектуры используется универсальный АТ-РЕШАТЕЛЬ [1], который позволяет сформировать фрагменты макета архитектуры на основе знаний о проблемной области.

Компонент конфигурации. Производится управление конфигурацией, которая определяет выбор используемого PDDL-планировщика, директорию для промежуточных файлов и т.д. Конфигурации хранятся в виде XML-файлов, а компонент содержит графическую оболочку для их редактирования конфигураций.

С помощью компонента интеграции с различными ПИК осуществляется передача интерактивных задач разработки в средства интерфейса комплекса АТ-ТЕХНОЛОГИЯ, а также получение результатов взаимодействия с инженером по знаниям. Кроме того, реализуется взаимодействие с ПИК комплекса АТ-ТЕХНОЛОГИЯ путем приема/передачи сообщений, а также выполняется ряд внутренних функций обработки состояния проекта разработки прототипа ИЭС с целью выделения определенных характеристик проекта и осуществляется контроль выполнения интерактивных задач плана (посредством передачи сообщений ПИК).

В процессе синтеза макета архитектуры АТ-РЕШАТЕЛЬ передает компоненту интеграции с различными ПИК инструкции по выполнению определенного рода задач. Для описания инструкций применяется специально разработанный язык, представляющий собой подмножество

XML, поскольку в качестве формата представления сообщений, используемых компонентами комплекса АТ-ТЕХНОЛОГИЯ, выбран XML.

Таким образом, использование интеллектуального планирования в управлении процессом построения прототипов ИЭС на основе задачно-ориентированной методологии и комплекса АТ-ТЕХНОЛОГИЯ [1] позволяет реализовать следующую функциональность: построение и реализацию плана разработки прототипа ИЭС на всех этапах ЖЦ с помощью интеллектуального планировщика; динамическое ассистирование инженеру по знаниям при построении текущего прототипа ИЭС на основе знаний о ТПП и ПИК, синтез архитектуры прототипа ИЭС и его компонентов на основе расширенной информационно-логической модели архитектуры ИЭС; анализ прототипа ИЭС за счет использования знаний о моделях и методах решения типовых задач; выдача рекомендаций и объяснений инженеру по знаниям. Соответственно, интеллектуальный планировщик «знает», сколько и каких ТПП и ПИКов зарегистрировано в комплексе и для чего они предназначены, а также какой алгоритм планирования используется, в соответствии с чем формируется набор задач для разработки любого прототипа ИЭС.

3.2. Типовые проектные процедуры и повторно-используемые компоненты

Примеры базовых ТПП детально рассматриваются в [1], в соответствии с чем все ТПП классифицируются следующим образом: ТПП, не зависящие от типа задачи (например, приобретение знаний из БД и др.), ТПП, зависящие от типа задачи (например, формирование компонентов обучающих ИЭС и др.), ТПП, связанные с ПИК, т.е. процедуры, содержащие информацию о жизненном цикле ПИК от начала его настройки до включения в макет прототипа, а также сведения о решаемых этим ПИК задачах и необходимых настройках.

В базовой версии комплекса АТ-ТЕХНОЛОГИЯ в настоящее время реализованы и используются несколько ТПП первого и второго типа, в частности: ТПП поддержки комбинированного метода приобретения знаний, ТПП проектирования баз данных средствами комплекса, ТПП конфигурирования компонентов ИЭС, ТПП создания гипертекстовых докумен-

тов и др. В стадии экспериментального программного моделирования находится ТПП, связанная с распределенным приобретением знаний из источников знаний различной типологии, ТПП, поддерживающая процессы построения динамических ИЭС, и достаточно сложная ТПП, реализующая построение обучающихся ИЭС.

В настоящее время в базовой версии комплекса АТ-ТЕХНОЛОГИЯ используются две группы ПИК – компоненты, реализующие возможности процедурного ПИК, и компоненты, реализующие возможности информационного ПИК. В первом случае, компоненты предоставляют возможности по выполнению некоторых действий, направленных на получение нетиповых результатов, т.е. тех действий, которые ранее не были накоплены в некотором хранилище (репозитории) или действий, требующих интерактивной работы с пользователем (например, просмотр протоколов интервьюирования экспертов и т.п.). Во втором случае компоненты обеспечивают возможности по выполнению действий, целью которых является получение ранее накопленной в репозитории информации (знаний, данных, схем, структур и т.д.) с дальнейшим копированием ее в текущий проект или же с дальнейшей обработкой этой информации (например, использование ранее созданной ER-диаграммы или анализ типовой диаграммы).

Объем технологической БЗ комплекса АТ-ТЕХНОЛОГИЯ в настоящее время составляет порядка 120 правил, которые используются для реализации ТПП на всех этапах ЖЦ разработки ИЭС для проблемных областей, связанных с диагностикой и проектированием. Кроме того при разработке прототипов ИЭС используются около 70 ПИК, среди которых 20 реализуют возможности информационных ПИК на базе единого репозитория, а порядка 50 реализуют возможности процедурных ПИК.

Заключение

В настоящее время проводится экспериментальное программное исследование текущей версии интеллектуального планировщика при прототипировании обучающихся ИЭС по различным курсам/дисциплинам. В перспективе планируется расширение возможностей интеллектуального планировщика для использования в

условиях командной разработки прототипов ИЭС при ограниченных ресурсах. Кроме того, начат цикл исследований и разработок, связанных с применением интеллектуального планировщика и других компонентов интеллектуальной программной среды для построения программного обеспечения двух прикладных динамических ИЭС («Управление медицинскими силами и средствами при крупных дорожно-транспортных происшествиях» и «Управление ресурсами системы спутниковой связи между региональными центрами»).

Литература

1. Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. Монография. — М: Научтехлитиздат, 2008. — 482с.
2. Рыбина Г.В. Практическое применение задачно-ориентированной методологии построения интегрированных экспертных систем (обзор приложений в статических и динамических проблемных областях) // Приборы и системы. Управление, контроль, диагностика. 2011. №12. С. 10-28.
3. Рыбина Г.В. Инструментальные средства построения динамических интегрированных экспертных систем: развитие комплекса АТ-ТЕХНОЛОГИЯ // Искусственный интеллект и принятие решений. 2010. №1. С. 41-48.
4. Рыбина Г.В. Модели и методы реализации интеллектуальной технологии построения интегрированных экспертных систем. // Приборы и системы. Управление, контроль, диагностика. 2011. №10. С. 27-37.
5. Рыбина Г.В., Блохин Ю.М., Иващенко М.Г. Некоторые аспекты интеллектуальной технологии построения обучающихся интегрированных экспертных систем // Приборы и системы. Управление, контроль, диагностика. 2013. №4. С. 27-36.
6. Рыбина Г.В., Иващенко М.Г., Блохин Ю.М. Интеллектуальная технология построения интегрированных экспертных систем // Искусственный интеллект и принятие решений. 2011. №3. С. 48-75.
7. Рыбина Г.В., Иващенко М.Г., Чурдалев Е.В., Мальцев И.Ю. Интеллектуальные программные среды: современное состояние и перспективы // Информационно-измерительные и управляющие системы. 2011. №10. С. 22-30.
8. Rybina G.V., Blokhin Yu.M., Danyakin I.D. Models, methods and tools for intelligent program environment for integrated expert systems construction // Информационно-измерительные и управляющие системы. Information-measuring and Control Systems. 2014. №8. С. 4-16.
9. Осипов Г.С. Методы искусственного интеллекта. — М.: ФИЗМАТЛИТ, 2011. — 296 с.
10. Nau D. S. Current trends in automated planning // AI Magazine. 2007. Vol. 28, №4. P. 43-58.
11. Рассел С., Нордвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2006. — 1408 с.
12. Penberthy J. S., Weld D. S. UCPOP: A sound, complete, partial order planner for ADL // Proceedings of the 3rd In-

- ternational Conference on Principles of Knowledge Representation and Reasoning KR'92. 1992. P. 103–114.
13. Helmert M. A planning heuristic based on causal graph analysis // *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*. 2004. P. 161–170.
 14. Bryce D., Kambhampati S. A tutorial on planning graph based reachability heuristics // *AI Magazine*. 2007. Vol. 28, №1. P. 47–83.
 15. Richter S., Westphal, Helmert. LAMA 2008 and 2011 (planner abstract). In: *Seventh International Planning Competition (IPC 2011), Deterministic Part*, 2011. P. 50–54.
 16. Benton J., Coles A. J., Coles A. Temporal planning with preferences and time-dependent continuous costs // *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling (ICAPS 2012)*. 2012. P. 2–10.
 17. Hakan, L. S. and Michael Littman. PPDD1.0: The language for the probabilistic part of IPC-4 // *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS-2004)*. 2004. P. 1–4.
 18. Ghallab M., Nau D. S., Traverso P. *Automated planning - theory and practice*. — Elsevier, 2004. — 663p.
 19. D. S. Nau, T. Au, O. Ilghami et al. SHOP2: an HTN planning system // *Journal of Artificial Intelligence Research (JAIR)*. 2003. Vol. 20. P. 379–404.
 20. D. McDermott, M. Ghallab, A. Howe et al. PDDL - the planning domain definition language : Tech. Rep. TR-98-003 / Yale Center for Computational Vision and Control. 1998.
 21. Johan Van Benthem and Jan Bergstra. Logic of Transition Systems // *Journal of Logic, Language, Information* №3. 1995. P. 247–283.
 22. Y. Alkharaji, M. Wehrle, R. Mattmuller, M. Helmert. A stubborn set algorithm for optimal planning // *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track*. 2012. P. 891–892.
 23. Helmert M., Domshlak C. Landmarks, critical paths and abstractions: What's the difference anyway? // *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009*. 2009. P. 162–169.
 24. C. Fawcett, M. Helmert, H. Hoos et al. Fd-autotune: Domain-specific configuration using fast downward // *ICAPS 2011 Workshop on Planning and Learning*. — 2011. — PP. 13–17.
 25. Bonet B., Geffner H. Planning as heuristic search // *Artificial Intelligence*. 2001. Vol. 129, №1–2. P. 5–33.
 26. Haslum P., Geffner H. Admissible heuristics for optimal planning // *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*. 2000. P. 140–149.
 27. Helmert M., Haslum P., Hoffmann J. Explicit-state abstraction: A new method for generating heuristic functions // *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13–17, 2008*. — 2008. — PP. 1547–1550.
 28. Richter S., Helmert M., Westphal M. Landmarks revisited // *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008*. 2008. P. 975–982.
 29. Karpas E., Domshlak C. Cost-optimal planning with landmarks // *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*. 2009. P. 1728–1733.
 30. Backstrom C., Nebel B. Complexity results for SAS+ planning // *Computational Intelligence*. 1995. Vol. 11. P. 625–656.
 31. Bonet B., Helmert M. Strengthening landmark heuristics via hitting sets // *ECAI 2010 - 19th European Conference on Artificial Intelligence*. 2010. P. 329–334.
 32. Hoffmann J., Nebel B. The FF planning system: Fast plan generation through heuristic search // *Journal of Artificial Intelligence Research (JAIR)*. 2001. Vol. 14. P. 253–302.
 33. Roger G., Helmert M. The more, the merrier: Combining heuristic estimators for satisficing planning // *Proceedings of the 20th International Conference on Automated Planning and Scheduling, ICAPS 2010*. 2010. P. 246–249.
 34. Helmert M., Geffner H. Unifying the causal graph and additive heuristics // *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008*. 2008. P. 140–147.
 35. Рыбина Г. В., Блохин Ю. М. Некоторые аспекты управления процессами построения интегрированных экспертных систем на основе интеллектуального планирования // *Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2014: Труды конференции*. Т.2. - Казань: Изд-во РИЦ Школа, 2014. С. 122–130.
 36. Яковлев К. С. HGA*: эффективный алгоритм планирования траектории на плоскости // *Искусственный интеллект и принятие решений*. 2010. №2. С. 16–25.
 37. Яковлев К. С., Баскин Е. С. Графовые модели в задаче планирования траектории на плоскости // *Искусственный интеллект и принятие решений*. 2013. №1. С. 5–12.
 38. G. Zou, Y. Chen, Y. Xu et al. Towards automated choreographing of web services using planning // *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012. P. 178–184.
 39. Garrido A., Morales L., Serina I. Using AI planning to enhance E-Learning processes // *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012*. 2012. P. 47–55.
 40. S. Parkinson, A. Longstaff, A. Crampton, P. Gregory. The application of automated planning to machine tool calibration // *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012*. 2012. P. 216–224.
 41. M. B. Do, L. Lee, R. Zhou et al. Online planning to control a packaging infeed system // *Proceedings of the Twenty-Third Conference on Innovative Applications of Artificial Intelligence*. 2011. P. 1636–1641.
 42. Beaudry E., Kabanza F., Michaud F. Planning with concurrency under resources and time uncertainty // *ECAI 2010 - 19th European Conference on Artificial Intelligence, Proceedings*. 2010. P. 217–222.
 43. И. Гофман, Д. Инишев, А. Липатов и др. Актуальные задачи развития технологии недоопределенного календарного планирования Time-Ex // *Одиннадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2008: Труды конференции*. Т.1. — М.: ЛЕНАНД, 2008. С. 68–76.
 44. Coles A. J. Opportunistic branched plans to maximise utility in the presence of resource uncertainty // *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track*. 2012. P. 252–257.

45. Бурков В. Н., Новиков Д. А. Как управлять проектами: Научно-практическое издание. — М. : СИНТЕГ-ГЕО, 1997. — 188 с.
46. Бурков В. Н., Коргин Н. А., Новиков Д. А. Введение в теорию управления организационными системами / Под ред. чл.-корр. РАН Д.А. Новикова. — М.: Либроком, 2009. — 264 с.
47. Гладков Л., Александров М. Решение задачи оперативного планирования производства на основе модифицированного генетического алгоритма с использованием нечеткого логического контроллера // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012: Труды конференции. Т.2. — Белгород: Изд-во БГТУ, 2012. С. 330-337.
48. J. Kietz, F. Serban, A. Bernstein, S. Fischer. Designing KDD-workflows via HTN-planning // ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track. 2012. P. 1011–1012.
49. Rybina G. V., Blokhin Y. M. Distributed knowledge acquisition control with use of the intelligent program environment of the AT-TECHNOLOGY workbench // Knowledge Engineering and the Semantic Web - 5th International Conference, KESW 2014. Proceedings. 2014. P. 150–159.
50. Колесов Н.В., Толмачева М.В., Юхта П.В. Оценки длительности вычислений и сложности их планирования в распределенных системах // Материалы 4-й Всероссийской мультikonференции. Т. 1. Таганрог: Изд-во ТТИ ЮФУ, 2011. С. 312-314.
51. A. Gerevini, P. Haslum, D. Long et al. Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners // Artificial Intelligence 2009. Vol. 173, №5-6. P. 619–668.
52. Новосельцев В., Пинжин А. Реализация эффективного алгоритма синтеза линейных функциональных программ // Известия Томского политехнического университета. 2008. Т. 312, №5. С. 32–35.

Рыбина Галина Валентиновна. Профессор кафедры «Кибернетика» Национального исследовательского ядерного университета «МИФИ» (НИЯУ МИФИ). Окончила МИФИ в 1971 году. Доктор технических наук, профессор, лауреат Премии Президента РФ в области образования. Автор более 450 печатных работ. Область научных интересов: интеллектуальные системы и технологии, интегрированные экспертные системы (статические, динамические, обучающие), интеллектуальные диалоговые системы, многоагентные системы, инструментальные средства.

Блохин Юрий Михайлович. Ассистент кафедры «Кибернетика» Национального исследовательского ядерного университета «МИФИ» (НИЯУ МИФИ). Окончил НИЯУ МИФИ в 2011 году. Автор около 20 печатных работ. Область научных интересов: интеллектуальные системы и технологии, интеллектуальное планирование, интегрированные экспертные системы (статические, динамические обучающие), эвристический поиск, инструментальные средства.