UNIFIED INTELLIGENT PLANNING FRAMEWORK: METHODOLOGY, FORMAL MODEL AND SEMANTIC TRACEABILITY

BORISLAV T. BORISOV

ABSTRACT: *The paper presents the methodology and formal model of the Unified Intelligent Planning Framework (UIPF), developed for domain-independent representation of scheduling problems. UIPF separates domain semantics, the unified formal core and optimization algorithms through bidirectional adapters and a stable planning interface. The paper summarizes the transformation pipeline, the canonical UIPF instance, semantic traceability and key correctness conditions. Production scheduling and university timetabling are used as two validation profiles with different semantics but a common formal contract. The scope of the conclusions is limited to the investigated ERP/UIS configurations and does not imply universal equivalence between scheduling domains.*

KEYWORDS: *intelligent scheduling, domain transformation, formal model, bidirectional adapter, semantic traceability, resource optimization*

DOI: https://doi.org/10.46687/PYMV8860

УНИФИЦИРАНА РАМКА ЗА ИНТЕЛИГЕНТНО ПЛАНИРАНЕ: МЕТОДОЛОГИЯ, ФОРМАЛЕН МОДЕЛ И СЕМАНТИЧНА ПРОСЛЕДИМОСТ

БОРИСЛАВ Т. БОРИСОВ

АБСТРАКТ: *В доклада се представят методологията и формалният модел на Унифицираната рамка за интелигентно планиране (UIPF), разработена за домейн-независимо представяне на задачи за планиране. Рамката разделя предметната семантика, унифицираното формално ядро и оптимизационните алгоритми чрез двупосочни адаптери и стабилен интерфейс към планировчика. Обобщени са конвейерът за преобразуване, каноничната UIPF инстанция, семантичната проследимост и основните условия за коректност. Производственото планиране и университетското разписание се използват като два валидационни профила с различна семантика, но общ формален контракт. Изводите са ограничени до изследваните конфигурации на ERP/UIS системи и не предполагат универсална еквивалентност между различните области на планиране*

1 Въведение

Задачите за оперативно планиране възникват в широк кръг организационни системи и обикновено изискват разполагане на дейности във времето, назначаване на ограничени ресурси и спазване на множество технологични, календарни и организационни правила. Класическата теория на scheduling предоставя устойчив формален апарат за такива задачи, включително модели за едномашинно, паралелно, job-shop, flow-shop и ресурсно ограничено планиране [1–3]. На практическо ниво обаче конкретната постановка често остава тясно свързана със структурата на изходната информационна система, с конкретния оптимизатор и с начина, по който резултатът се връща за изпълнение.

При динамична среда този проблем се усложнява допълнително, защото нови поръчки, промени в наличността на ресурси, откази, календарни изменения и фактически данни от изпълнението налагат актуализиране или препланиране на вече изграден график. Литературата за производствено rescheduling разглежда стратегии и политики за реакция

при такива събития [4], но самото препланиране не решава въпроса как една и съща архитектура да работи устойчиво върху различни домейнови модели и различни алгоритмични реализации.

Изследователският проблем, разглеждан тук, е по-високо ниво на абстракция: как предметната специфика да бъде запазена и проследима, без тя да прониква в самото формално ядро и във вътрешната кодировка на оптимизационния метод. Търси се рамка, при която производствена операция и учебно занятие могат да бъдат разглеждани като различни специализации на общата категория „дейност“, а машина, оператор, преподавател, студентска група и аудитория — като различни типове ресурси, без да се заличават съществените различия между предметните области.

В отговор се предлага Унифицирана рамка за интелигентно планиране (Unified Intelligent Planning Framework — UIPF). Централната идея е да се разграничат: домейновото описание; техническата интеграция; семантичният адаптер; унифицираното формално ядро; алгоритмичният слой; обратното интерпретиране на решението; и валидацията. Докладът концентрира изложението върху методологията за преобразуване, каноничния формален модел, семантичната карта и условията, при които един домейнов адаптер може да се счита за коректен. Алгоритмите се разглеждат като заменяеми реализации, а не като дефиниция на самата рамка.

2 Методологична постановка и дефиниция на UIPF

Методологично UIPF се позиционира между предметния модел и оптимизационната постановка. Разделянето на изходен модел, трансформация и целеви модел е съвместимо с общите идеи на model transformation [5, 6], но тук те са конкретизирани за задачи за планиране чрез двупосочни домейнови адаптери, унифицирана scheduling схема и задължително обратно интерпретиране на резултата. Двупосочните преобразувания са добре изследван проблем при поддържане на съгласуваност между различни представяния [7, 8]; в UIPF тази идея се използва като архитектурен принцип за връзката между предметните обекти, унифицирания модел и крайния график.

UIPF се дефинира като домейн-независима концептуална, методологична, формална и архитектурна рамка за преобразуване, моделиране, оптимизиране, изпълнение и оценяване на задачи за разпределение на дейности и ограничени ресурси във времето. Съществено е, че рамката не цели да премахне предметната семантика. Тя я локализира в адаптера и семантичната карта, така че оптимизационният слой да работи с общи категории, а резултатът да остава възстановим и проверим в термините на изходната система.

$$\mathrm{UIPF} = \mathrm{M_D} + \mathrm{U} + \mathrm{O} + \mathrm{A} + \mathrm{V} \tag{1}$$

В (1) M_D означава методологията за домейново преобразуване; U — унифицираното формално ядро; O — общия интерфейс към методите за планиране и оптимизация; A — архитектурата за интеграция, изпълнение и обратна връзка; а V — методиката за валидация и оценяване. Знакът „+“ обозначава концептуално обединяване, а не аритметична операция. Формален модел без адаптер не е достатъчен за надеждна връзка с реална информационна среда; адаптер без общо ядро остава специфична интеграция; оптимизатор без стабилен входно-изходен контракт не е заменяем компонент.

Проследимостта е самостоятелно изискване, а не страничен ефект от преобразуването. При пренасяне на реални данни е необходимо да се съхранява информация не само кой домейнов идентификатор на кой унифициран обект съответства, но и по какво правило е създадена връзката, коя версия на входното състояние е

използвана и дали е извършена декомпозиция, агрегиране или извеждане на нов параметър. Тази постановка е близка до общите принципи за data provenance [9, 10], но в UIPF произходът е обвързан пряко с жизнения цикъл на scheduling решението.

3 Архитектура и методология за домейново преобразуване

Архитектурата на UIPF е многослойна, двупосочна и разширяема. Външните ERP, MES, UIS, IoT и други системи предоставят данни и събития, но не са част от формалното ядро. Между тях и ядрото са разположени техническият интеграционен слой и домейновият адаптер. Това разделение е важно: извличането на записи от база данни, синхронизацията и конвертирането на формати са различен проблем от решението, че даден запис представлява дейност, ресурс, ограничение или параметър на целева функция.



Фигура 1. Вътрешна многослойна архитектура на Унифицираната рамка за интелигентно планиране (UIPF)

Шестте функционални слоя на фиг. 1 са: интеграция с предметните системи; домейново преобразуване; унифицирано ядро; интелигентно планиране и оптимизация; изпълнение и обратна връзка; валидация и оценяване. Архитектурата не предполага еднократно линейно изпълнение. След получаване на домейнов график резултатите и новите събития могат да актуализират следващото състояние, което затваря цикъла на планиране.

(2) $$D_raw \rightarrow[\Phi_D]\ U_valid(D,t) \rightarrow[\Omega_k]\ S(D,t) \rightarrow[\Psi_D]\ Y(D,t) \rightarrow[\Gamma_D]\ Dhat(D,t+1)$$

Оперативният поток в (2) започва от сурово или частично структурирано домейново описание. След техническо нормализиране се получава Dhat_D,t със съгласувани идентификатори, формати и времеви единици, но със запазена предметна семантика. Forward преобразуването Ф_D изгражда валидирана UIPF инстанция, оптимизаторът Ω_k

генерира решение в унифицираното пространство, Ψ_D го превежда към домейнов кандидатен график, а Γ_D обработва резултатите от изпълнението и новите събития. Така reverse mapping се различава от евентуалната deployment операция за физически запис в ERP или UIS; последната остава контролирано действие на интеграционния слой.

Таблица 1

Етапи на методологията за домейново преобразуване към UIPF

№	Етап	Основен резултат
1	Определяне на границите	Обхват, хоризонт и детайлност на задачата
2	Обекти и решения	Каталог на домейновите обекти и променливите на решението
3	Дейности	Множество от обекти, които трябва да бъдат разположени във времето
4	Ресурси	Типове ресурси, способности, квалификации и капацитети
5	Време и календари	Нормализиран времеви модел и наличност
6	Зависимости	Предшестване, синхронизация, групиране и взаимно изключване
7	Ограничения	Разделяне на твърди и меки ограничения
8	Цели и KPI	Целеви функции и показатели за качество
9	UIPF инстанция	Построяване, семантична карта и структурна валидация
10	Оптимизация	Избор и изпълнение на заменяем оптимизационен модул
11	Обратно преобразуване	Домейнов график, проверка и обратна връзка

Единадесетте етапа в табл. 1 са организирани в четири фази: анализ на предметната област; формално абстрахиране; построяване и валидиране на UIPF инстанцията; оптимизация, обратно преобразуване и обратна връзка. Последователността е методологична, а не задължително програмно линейна. Например преобразуването на ограниченията може да изисква връщане към ресурсните квалификации или календара. Основното изискване е резултатът да бъде пълен, проследим и проверим преди предаването му към алгоритмичния слой.

4 Унифицирано формално ядро

Централният теоретичен компонент на рамката е унифицираното ядро. То съдържа само категории, които имат смисъл независимо от конкретната предметна терминология. В ядрото няма SQL заявки, ERP таблици, хромозоми на генетичен алгоритъм, феромонни матрици, правила за визуализация или понятия като „машина“ и „аудитория“. Те принадлежат на интеграционния, адаптационния, алгоритмичния или изпълнителния слой. Това разделяне позволява една и съща семантична инстанция да бъде проектирана към различни вътрешни модели на оптимизатора.

(3) $$U(d,t) = < J, R, T, \Delta, \Lambda, P, Q, C_H, C_S, F, X_t, E_t, M_d >$$

Каноничната инстанция (3) включва: J — множество от дейности; R — ресурси; T — времеви модел и календар; Δ — продължителности и времеви характеристики; Λ — допустими назначения; P — зависимости; Q — изисквания, квалификации, способности и капацитети; C^H — твърди ограничения; C^s — меки ограничения; F — целеви функции; X_t — текущо състояние; E_t — нормализирани събития; M_d — семантична карта към изходните домейнови обекти. Производствените и университетските понятия не променят тази схема; те се преобразуват към различни стойности и типове в общите множества.

Дейността j_i съдържа идентификатор, унифициран тип, продължителност или функция за нейното определяне, времеви прозорец, ресурсни изисквания, приоритет, състояние и метаданни за проследимост. Ресурсите се класифицират като човешки, технически, пространствени, материални, групови и информационни. Класификацията не означава, че всяка задача използва всички типове; тя осигурява общ език за различни комбинации от назначения. Например в производствен профил една операция може да изисква машина и квалифициран оператор, а в университетски профил едно занятие — преподавател, студентска група и подходяща аудитория.

Множеството Λ описва допустимите назначения и отделя физически или организационно възможните комбинации от тези, които оптимизаторът трябва да разглежда. P представя отношенията между дейностите, включително предшестване и синхронизация. Q съдържа квалификации, способности и капацитети. Твърдите ограничения C^H определят допустимостта, докато меките ограничения C^s измерват предпочитания или качества, чието нарушение може да бъде санкционирано. Това разграничение е стандартно за много scheduling постановки [1–3], но в UIPF то се използва като част от домейн-независимия контракт.

Целевите функции F не са фиксирани до един показател. В зависимост от профила могат да се оценяват makespan, закъснение, използваемост, пренастройки, стабилност при препланиране, балансираност, паузи или предпочитания. При множество конфликтни цели рамката допуска както агрегиран скаларен критерий, така и Pareto постановка. Теорията на многокритериалната оптимизация предоставя формалната основа за такива задачи [11–13], а NSGA-II е представителен алгоритъм за еволюционно Pareto търсене [14]. В UIPF изборът между тези стратегии е част от алгоритмичната конфигурация, а не промяна на домейновия адаптер.

Динамичният компонент X_t съдържа състоянията на дейностите и ресурсите, текущите и фиксираните назначения, фактическите данни от изпълнението и версия на snapshot-а. E_t описва събития като добавяне или промяна на дейност, недостъпност на ресурс, календарна промяна, промяна на приоритет или отклонение при изпълнение. Събитието се нормализира към обща структура и се прилага върху X_t, след което се изгражда нова UIPF инстанция със същата канонична схема и актуализирани само засегнати компоненти.

Преди оптимизацията се изпълнява структурна валидация. Проверяват се уникалност на идентификаторите, положителни продължителности, непротиворечиви времеви прозорци, съществуване на потенциално допустим ресурсен набор за всяка задължителна дейност, валидност на препратките и наличие на семантичен произход за домейново получените елементи. Структурно валидна инстанция не гарантира непременно съществуване на допустим график; тя гарантира, че моделът е пълен и достатъчно непротиворечив, за да бъде предаден към математическата и алгоритмичната проверка.

5 Двупосочен адаптер и семантична проследимост

Домейновият адаптер е границата, в която предметната семантика се преобразува към общото ядро и се възстановява след оптимизацията. Forward частта Φ_d може да се разглежда като композиция от частични преобразувания за дейности, ресурси, време, допустими назначения, квалификации и капацитети, зависимости, ограничения, цели, състояние, събития и семантична карта. Така всяка група правила може да бъде проверявана отделно, вместо преобразуването да се третира като неделима операция.

(4) $$M_d \subseteq O_d \times E_U$$

В (4) O_d е множеството от релевантни домейнови обекти, а E_U — елементите на унифицираното ядро. На практика записът в M_d трябва да съдържа повече от двойка идентификатори: източник, цел, тип на преобразуването, роля, идентификатор на правило и версия. Типът на преобразуването може да бъде direct, decomposition, aggregation, derived, constraint или parameter. Това е необходимо, когато един домейнов обект се разделя на няколко дейности, когато няколко записи се агрегирани в един ресурс или когато формално ограничение се извежда от комбинация от атрибути.

След оптимизацията нормализираното решение съдържа за всяка планирана дейност назначен ресурсен набор, начален и краен момент и статус. Обратното преобразуване Ψ_d използва M_d, за да възстанови съответните домейнови идентификатори, ресурсни роли, времеви стойности и версия на графика. Полученият Y_d,t е кандидатен домейнов график. Преди използване той трябва да бъде валидиран спрямо актуалното домейново състояние, включително съществуване и наличност на ресурсите, спазване на твърдите правила и съвместимост с snapshot-a, върху който е извършена оптимизацията.

Това разграничение има практическо значение. Коректният reverse mapping не означава автоматично, че системата трябва да извърши директен запис в оперативна ERP или UIS база. Публикуването може да изисква одобрение, транзакционна проверка или допълнителни бизнес правила и следва да бъде отделна deployment операция. По този начин формалната коректност на адаптера не се смесва с организационната политика за внедряване.

6 Условия за коректност и алгоритмична независимост

Коректността на адаптера не може да се сведе до успешно прочитане на данните или до синтактично валиден изход. Необходими са поне: валиден и версиониран вход; семантично покритие на съществените обекти; типова съвместимост; референтна затвореност; запазване на твърдите ограничения; съгласувано измерване на меките ограничения и целите; пълна семантична карта; обратно интерпретируемо решение; запазване на домейновата допустимост; събитийна и версионна съгласуваност; алгоритмична независимост; и локалност на домейновите промени.

(5) $$\begin{aligned}&\mathrm{Correct}(\mathrm{DA_d}) \Rightarrow \mathrm{ValidCore}(\Phi_d(\mathrm{Dhat_valid})) \wedge \mathrm{Traceable}(\mathrm{M_d}) \\ &\wedge \mathrm{VersionConsistent}(U(d,t), Y(d,t)) \wedge [\, S \in S_feas(U(d,t)) \Rightarrow \mathrm{Feasible_d}(\Psi_d(S)) \,]\end{aligned}$$

Условие (5) синтезира минималния контракт: валидният вход трябва да се преобразува към валидно ядро; допустимо решение в унифицираното пространство не трябва да се превръща в недопустим домейнов график; семантичната карта трябва да осигурява проследимост; а решението трябва да остава съгласувано с използваната версия на състоянието. Пълно формално доказателство за всички възможни входове би изисквало формална спецификация на цялата домейнова схема и на всички трансформационни правила. В практическа изследователска реализация доказателствата се комбинират от формални условия, структурни тестове, експериментални сценарии и експертна проверка на предметната интерпретация.

Алгоритмичната независимост се реализира чрез отделен контракт между UIPF инстанцията и конкретния оптимизатор. Един метод може да използва хромозома, друг — феромонна структура, трети — математически модел или constraint представяне. Вътрешните им структури не се включват в U_d,t. Всеки оптимизатор получава семантично еквивалентна проекция на същата инстанция и връща резултат, който се нормализира към общ формат на решение. Така сравняването на алгоритми може да се

извършва при едни и същи ограничения и KPI, без домейновият адаптер да бъде пренаписван за всеки нов метод.

Този принцип е важен и при динамично препланиране. При събитие адаптерът определя засегнатите домейнови и UIPF обекти, а алгоритмичният слой избира политика като update-only, local repair, warm start или пълно препланиране. Семантичната интерпретация на събитието и изборът на метод остават разделени. Така рамката може да бъде разширявана с нов оптимизатор или нова стратегия за реакция, без да се изменя основният адаптерен контракт.

7 Междудомейнова интерпретация и граници на приложимост

Домейн-независимостта на UIPF не означава, че различните предметни области са еквивалентни. Тя означава, че съществените им scheduling конструкции могат да бъдат представени чрез една и съща формална схема, докато специфичните правила остават локализирани в адаптерите. Университетското разписание е подходящ втори профил, защото има утвърдена самостоятелна литература и специфични ограничения за преподаватели, студентски групи, помещения и времеви слотове [15–17]. Тези понятия не се заменят механично с производствени термини; те се проектират към общите категории J, R, T, Λ, Q, C^H, C^s и F.

В производствения профил дейностите са операции от технологични маршрути, техническите ресурси са машини, човешките ресурси са оператори, а целите могат да включват закъснение, makespan, пренастройки и използваемост. В университетския профил дейностите са учебни занятия, пространствените ресурси са аудитории, човешките — преподаватели, а груповите — студентски групи или потоци; типични цели са минимизиране на конфликти, паузи и нарушения на предпочитанията. Общото е формата на scheduling решението, не предметното значение на обектите.

Практическата мотивация на производствения профил се развива последователно в предходни изследвания на автора: генетичен алгоритъм за дискретно производствено планиране [18], ACO за обработка на поръчки и автоматизация в ERP–MES–IoT среда [19] и SMA за планиране на екструзионна линия с ERP-базиран агент [20]. В UIPF тези алгоритмични и интеграционни разработки се третират като специализирани реализации и емпирична основа, върху която се формулира общият адаптерен и формален контракт.

Границата на извода е съществена. Наличието на работещ производствен и университетски адаптер подкрепя домейн-независимостта в рамките на изследваните профили, но не доказва, че всяка задача за планиране може безусловно да бъде представена от текущата схема. При нов домейн трябва да се провери дали неговите съществени дейности, ресурси, времеви характеристики, зависимости, ограничения и цели са представими. Ако е необходима нова конструкция, разширението следва да бъде формулирано на общо ниво, а не чрез понятие, валидно само за един частен случай.

8 Заключение

Представената методология разглежда интелигентното планиране като проследим двупосочен процес от домейновите данни към унифицирано формално ядро, алгоритмично решение и обратно интерпретируем график. Основният принос на UIPF е в разделянето на предметната семантика, формалния модел и оптимизационния алгоритъм чрез стабилен адаптерен контракт. Каноничната инстанция обединява дейности, ресурси,

време, допустими назначения, зависимости, изисквания, твърди и меки ограничения, цели, състояние, събития и семантична карта.

Единадесетстъпковият Transformation Pipeline осигурява възпроизводима последователност за въвеждане на нов домейн, а условията за коректност поставят акцент върху семантичното покритие, запазването на допустимостта, проследимостта и версионната съгласуваност. Производственото планиране и университетското разписание показват как различни предметни модели могат да използват общото ядро при запазване на специфичните си ограничения. Следващата естествена посока е разширяване на набора от домейнови адаптери и систематично сравняване на алгоритми върху еднакви UIPF инстанции, без промяна на семантичния слой.

ЛИТЕРАТУРА:

[1] Pinedo, M., Scheduling: Theory, Algorithms, and Systems. 5th ed., Springer (2016).

[2] Brucker, P., Scheduling Algorithms. 5th ed., Springer, Berlin–Heidelberg (2007).

[3] Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., Weglarz, J., Handbook on Scheduling. Springer (2007).

[4] Vieira, G.E., Herrmann, J.W., Lin, E., Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. Journal of Scheduling, 6 (1) (2003), 39–62.

[5] Mens, T., Van Gorp, P., A Taxonomy of Model Transformation. Electronic Notes in Theoretical Computer Science, 152 (2006), 125–142.

[6] Czarnecki, K., Helsen, S., Feature-Based Survey of Model Transformation Approaches. IBM Systems Journal, 45 (3) (2006), 621–645.

[7] Foster, J.N., Greenwald, M.B., Moore, J.T., Pierce, B.C., Schmitt, A., Combinators for Bidirectional Tree Transformations: A Linguistic Approach to the View-Update Problem. ACM Transactions on Programming Languages and Systems, 29 (3) (2007), 17.

[8] Stevens, P., Maintaining Consistency in Networks of Models: Bidirectional Transformations in the Large. Software and Systems Modeling, 19 (2020), 39–65.

[9] Simmhan, Y.L., Plale, B., Gannon, D., A Survey of Data Provenance in e-Science. ACM SIGMOD Record, 34 (3) (2005), 31–36.

[10] Herschel, M., Diestelkämper, R., Ben Lahmar, H., A Survey on Provenance: What For? What Form? What From? The VLDB Journal, 26 (6) (2017), 881–906.

[11] Miettinen, K., Nonlinear Multiobjective Optimization. Springer, Boston (1999).

[12] Ehrgott, M., Multicriteria Optimization. 2nd ed., Springer, Berlin–Heidelberg (2005).

[13] Deb, K., Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001).

[14] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6 (2) (2002), 182–197.

[15] Schaerf, A., A Survey of Automated Timetabling. Artificial Intelligence Review, 13 (2) (1999), 87–127.

[16] Lewis, R., A Survey of Metaheuristic-Based Techniques for University Timetabling Problems. OR Spectrum, 30 (1) (2008), 167–190.

[17] Ceschia, S., Di Gaspero, L., Schaerf, A., Educational Timetabling: Problems, Benchmarks, and State-of-the-Art Results. European Journal of Operational Research, 308 (1) (2023), 1–18.

[18] Borisov, B., Optimization of Production Planning for Discrete Manufacturing Using a Genetic Algorithm. Proceedings of the 10th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE) (2025), 1–7.

[19] Borisov, B., Todorov, V., Ant Colony Optimization for Order Processing and Labeling Automation in Food Manufacturing Using ERP–MES–IoT Integration. TEM Journal, 15 (2) (2026), 960–969.

[20] Borisov, B., Order Planning Optimization for Composite Film Extrusion Using Slime Mold Optimization Algorithm and an ERP-Based Agent. Proceedings of the 11th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE), Starozagorski Bani, Bulgaria (2026).

Борислав Т. Борисов
Шуменски университет „Епископ Константин Преславски“, Факултет по математика и информатика,
Катедра „Компютърни системи и технологии“
ORCID ID: 0009-0002-8105-9815
E-mail: b.t.borisov@shu.bg