



УДК 004.041:004.042

## CONCEPTION OF INFORMATION TECHNOLOGY OF REPRESENTATION OF COMPUTATIONAL PROCESS BY PETRY NET КОНЦЕПЦИЯ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ПРЕДСТАВЛЕНИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА СЕТЬЮ ПЕТРИ

N.O. Komlevaya/ Н. О. Комлевая

c.t.s, as.prof/к.т.н., доц.

ORCID: <https://orcid.org/0000-0001-9627-8530>

O.N. Paulin/ О. Н. Паулин

d.t.s, as.prof/д.т.н., доц.

ORCID: <https://orcid.org/0000-0002-2210-8317>

Odessa State University "Odessa polytechnica" Odessa, av. Shevchenko, 1

Одесский государственный университет «Одесская политехника»,

Одесса, пр. Шевченко, 1

**Аннотация.** Целью работы является повышение качества программных продуктов путем представления и оптимизации вычислительных процессов (ВП) сетью Петри.

Рассматривается новый подход к анализу программ, основанный на предварительном представлении ВП сетями Петри (СП). Это позволит выявить множество ошибок уже на стадии отладки ВП, а также оптимизировать ВП.

В статье предлагается концепция информационной технологии (ИТ) построения ВП на основе его представления СП. Исходной информацией для ИТ является словесное описание алгоритма решения задачи, выходной информацией является качественный ВП и соответствующая ему СП.

ИТ включает в себя следующие этапы: построение исходного ВП и итерационное повышение его качества путём устранения распространённых ошибок и оптимизации; декомпозиция ВП на макрооперации и составление для них фрагментов СП; моделирование фрагментов СП; составление полной СП и её моделирование; занесение в библиотеку МО и и отлаженного полного ВП, а также соответствующих им фрагментов СП и полной СП.

Описывается содержание этапов ИТ с учётом достигнутых результатов.

**Ключевые слова:** качество программного продукта, концепция, информационная технология, этап, словесное описание алгоритма, вычислительный процесс, декомпозиция, фрагмент, сеть Петри, моделирование, оптимизация, библиотека.

### Вступление

При разработке программного продукта (ПП) на основе современных технологий необходимо соблюдение требований, выраженных стандартами (ISO 12207, ISO 9000, СММ). Однако этого мало: практика полна примеров больших затрат времени и ресурсов на исправление ошибок и реинжиниринг в будущем. Наиболее критичными являются этапы программирования и тестирования, которые непосредственно влияют на качество ПП (ISO 9001). Поэтому качественное программирование и тестирование программ позволяет значительно сокращать время на разработку и повышает качество ПП в целом, что очень актуально.

Рост сложности программ приводит к увеличению времени их отладки и тестирования, причем анализ программ часто занимает больше времени, чем написание самой программы. На наш взгляд, это следствие неправильной стратегии создания ПП, так как ошибки надо выявлять на более ранней стадии решения некоторой задачи – на стадии алгоритмизации, то есть разработки



вычислительного процесса (ВП). На этой же стадии следует заниматься и оптимизацией ВП, что повысит качество программы.

Данная статья является дальнейшим развитием работы [1], в которой предлагается новый подход к повышению качества ПП за счёт предварительного повышения качества ВП. Здесь рассмотрены: современные приложения теории сетей Петри к разнообразным ВП: характеристики сетей Петри и их оптимизация; применение СП в случае более сложных многоуровневых ВП. Однако авторами не сделаны попытки уточнения ВП на основании результатов моделирования соответствующих СП.

### **Основная часть**

**Постановка проблемы.** Процесс получения качественного ПП включает в себя основные этапы его разработки [2]: формулирование идеи решения задачи; развертывание идеи в словесное описание ВП; доказательство того, что ВП является алгоритмом; построение схемы алгоритма; написание программ на языке высокого уровня; отладка программы на контрольном примере и тестирование.

Однако на практике обычно делается акцент на этап написания программы; при этом предыдущие этапы (кроме первого) или выпускаются, или выполняются частично и небрежно. Это приводит к необходимости последующего рефакторинга программного кода. Такая «традиция» имеет следующие недостатки:

1. Ошибки в процедуре уже не могут быть исправлены программно, поэтому важно их обнаружить на этапе анализа схемы алгоритма.

2. Многообразие вычислительных процессов при традиционном подходе приводит к излишним затратам времени на программирование каждого отдельного ВП и на анализ программ, в то время как в вычислительных процессах есть много общего в виде подпроцессов.

Известны свободные интегрированные среды разработки (IDE) программного обеспечения (ПО) для различных языков программирования, такие как IntelliJ IDEA, NetBeans, Eclipse и др. Они решают проблему ускорения создания ПО с использованием объектно-ориентированного подхода, но используются на последнем этапе разработки ПО.

Существуют различные CASE-технологии (Computer Aided Software Engineering), среди которых наиболее известными являются ERwin, PowerBuilder, SQL Windows, DataBase Designer и др. Кроме того, существуют специальные языки моделирования, наиболее распространенным из которых является UML – Unified Modeling Language. Эти инструменты работают с графическими описаниями архитектурных и функциональных элементов в области разработки ПО, моделирования бизнес-процессов, проектирования систем и отображения организационных структур.

Функции CASE-технологий включают инструменты для анализа, проектирования и программирования ПО, проектирования интерфейсов, документирования и создания структурированного кода на языке программирования. Однако часто эти инструменты являются средствами объектного моделирования, нацеленными на создание абстрактной модели



системы и не учитывающими качество ВП.

В последнее время популярным стал автоматный подход к программированию, в котором можно выделить два наиболее развитых подхода: SWITCH-технология [3] и КА-технология [4] (КА – конечный автомат) разработки ПО. Эти два подхода различаются реализацией логики автоматных программ. Однако недостатком автоматного подхода является малая вычислительная мощность, выражаемая автоматным языком. Более адекватным является использование сетей Петри [5], которые имеют существенно *большую* выразительность своего языка, занимая промежуточное положение между КА и машиной Тьюринга. К тому же автомат не отражает более сложного понятия, чем «операция», т.е. понятие «событие», а также взаимную зависимость «событие – условие», что характерно для сетей Петри; это сужает сферу применения КА.

При автоматном подходе программные конструкции преобразуются в базовые автоматы, что также является недостатком, так как целесообразно связывать с автоматом вычислительные конструкции (МО). Это тем более относится к сетям Петри.

В свете сказанного особенно важной и актуальной является разработка информационной технологии автоматизированного построения сети Петри для заданного ВП. Решению данной проблемы, а также обсуждению связанных с этим предпосылок и анализа результатов, и посвящена данная работа.

**Цели и задачи исследования.** Целью работы является повышение качества ПП и, в конечном счёте, сокращение времени на анализ программы, реализующей конкретный ВП, за счет более ранней увязки вычислительных конструкций с более мощным средством их описания – сетью Петри.

Для достижения этой цели необходима разработка информационной технологии (ИТ) представления ВП сетью Петри. При этом должны быть решены следующие основные задачи:

- построение библиотеки ВП и соответствующих им СП в виде базы знаний;

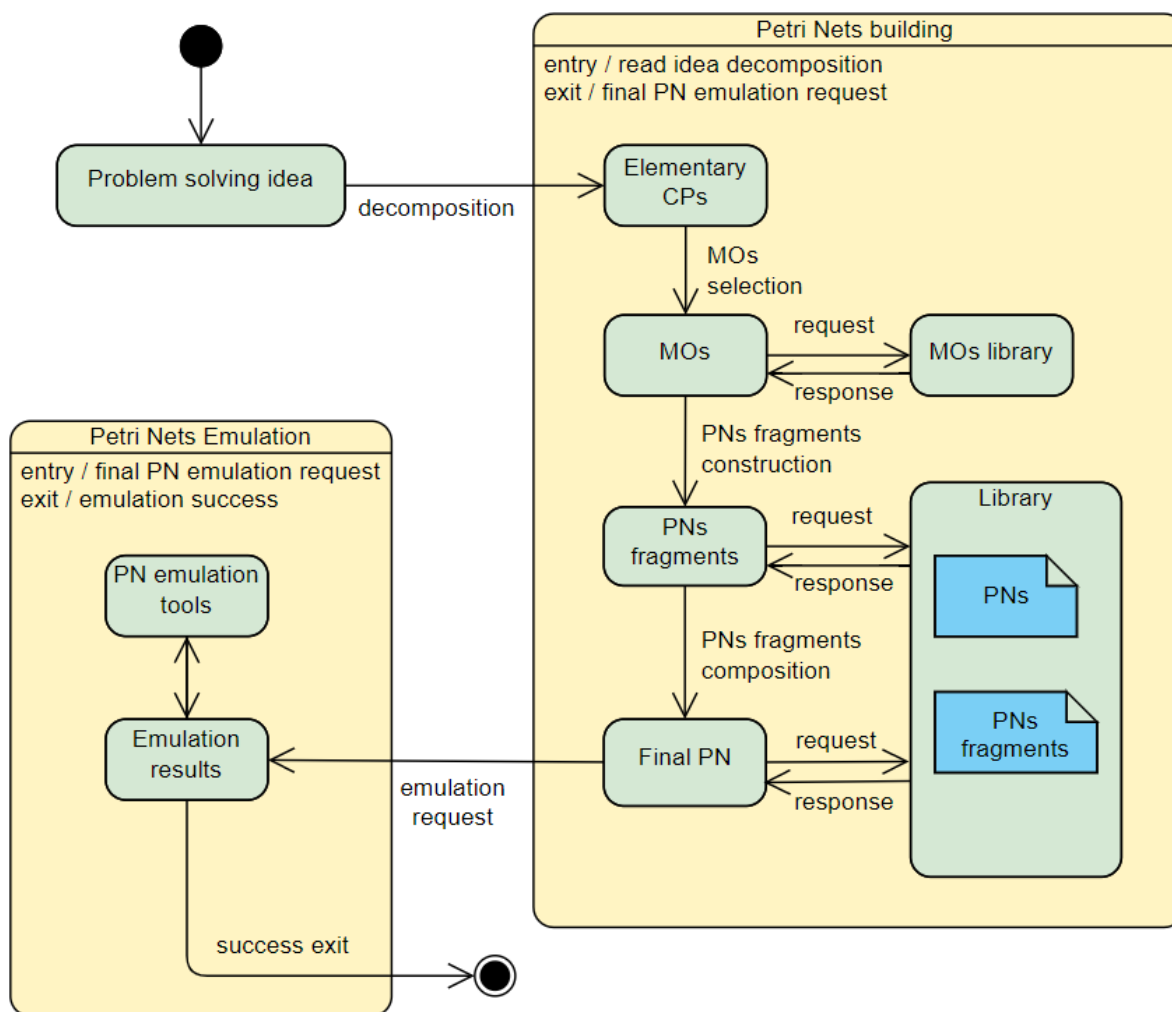
- разработка методов и методик для: 1) декомпозиции ВП на МО (вычислительные и управляющие – см. [1]); 2) преобразования МО во фрагменты СП; 3) композиции фрагментов в полную СП; 4) моделирования фрагментов и полной СП; 5) описания МО и финального ВП и соответствующих им фрагментов и полной СП; 6) реализации операций над библиотекой (занесение, извлечение, редактирование описаний), которая по сути должна представлять собой базу знаний.

В работе представлено частичное решение указанных задач.

### **Концепция информационной технологии.**

В [1] предложены метод и реализующая его процедура построения модели ВП с использованием СП, что позволило разработать информационную технологию преобразования ВП в СП.

На рис. 1 представлена информационная технология (ИТ) разработки ВП с использованием сети Петри. Здесь CPs – computational processes; MOs – macro-operations; PN – Petri net.



**Рисунок 1 – ИТ разработки ВП с использованием сетей Петри**

На первом этапе ИТ выполняется декомпозиция решения поставленной задачи на элементарные вычислительные процессы (подпроцессы).

Далее каждый такой подпроцесс оформляется в виде МО. Если процесс рассматривается впервые, для него вручную выделяются МО и они заносятся в библиотеку. Если же подпроцесс встречается повторно, данный этап не выполняется.

Каждой выделенной МО сопоставляется фрагмент сети Петри, который после моделирования также заносится в библиотеку.

Композиция фрагментов сетей Петри формирует финальную сеть Петри, которая заносится в библиотеку наряду с соответствующим ВП.

**Построение сети Петри по выделенным МО.**

Исходим из того, что ВП включает в себя две составляющие: вычислительную и управляющую. Первая составляющая использует совокупность операций разного уровня иерархии, начиная с простейших операций процесса вычислений; вторая составляющая обеспечивает организацию управления вычислительным процессом, то есть выстраивание процесса в определенном порядке.

Предлагаемый нами подход основан на следующих положениях.



**Утверждение.** Любой ВП может быть сконструирован на основе относительно небольшого числа МО и структур управления.

**Определение 1.** *Вычислительный процесс* – это последовательность операций как простых, так и достаточно сложных (макроопераций и процедур), выстроенных во времени в определённом порядке.

**Определение 2.** *Макрооперацией* называется логически законченный фрагмент вычислительного процесса.

МО включает в себя на нижнем уровне иерархии простые арифметические операции и арифметические выражения (как и в программировании, при рассмотрении ВП следует различать равенство и присваивание), на следующем уровне – перестановки элементов массива, смещения элементов последовательности, сравнение строк (столбцов) таблицы и т.п. В МО могут быть встроены структуры управления (альтернатива, итерация и более сложные структуры) элементарными операциями и МО. Так, простую сортировку вставками можно рассматривать как МО 3-го уровня иерархии с двумя циклами, управляющими процессом сортировки.

### **Библиотека.**

Структура разрабатываемой библиотеки строится по принципам обычной библиотеки. Например, разделам соответствуют определённые классы решаемых задач (сортировка, покрытия, на графах и т. д.), которые включают в себя конкретные алгоритмы. Компонентами библиотеки являются МО и полный ВП, а также соответствующие им фрагменты и полная СП. Основные операции над компонентами библиотеки – это занесение компонента в библиотеку, его вывод, редактирование

Также имеются алгоритмы, используемые в разных классах задач. Такого рода общие алгоритмы должны быть выделены в отдельную группу.

В определенном разделе библиотеки должен быть приведен минимизированный перечень МО для всех алгоритмов данного класса задач с подробным описанием сути МО, её спецификацией.

### **Заключение и выводы.**

Предложенная ИТ основана на материале [1]. Идея перехода к более крупным алгоритмическим фрагментам основана на утверждении о возможности описания большого разнообразия ВП с помощью относительно небольшого числа МО, которым соответствуют отлаженные фрагменты СП.

ИТ построения сети Петри для данного ВП требует дальнейшей проработки в части методики преобразования МО в фрагменты СП, библиотеки как базы знаний, методики моделирования СП.

### **Литература**

1. Paulin, O. N., Komleva, N. O., Marulin, S. U. & Nikolenko, A. A. “Method for Constructing the Model of Computing Process Based on Petri Net”. *Applied Aspects of Information Technology. Publ. Science i Technical*. Odessa: Ukraine. 2019; Vol. 2 No.4: 260–270. DOI: <https://doi.org/10.15276/aait.04.2019.1>

2. Goodman, S.E. and Hedetniemi, S.T. Introduction to the Design and Analysis of Algorithms [Text] Tata McGraw-Hill, New Delhi, 2002.



3. Шалыто А. А. SWITCH-технология. Алгоритмизация и программирование логических задач управления [Текст] – СПб: Наука, 1998. – 628 с.
4. Бурдонов И. Б., Косачев А. С., Кулямин В. В. Использование конечных автоматов для тестирования программ [Текст] // Программирование. – 2000. – №2. – С. 12-28.
5. Girault C., Valk R. Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications [Online] – Springer, 2003. — 601 p.

**Abstract.** *The aim of the work is to improve the quality of software products by preliminary representation of computational processes (CP) by a powerful tool for their description - Petri net (PN). This will allow you to identify many errors already at the stage of debugging the CP, as well as optimize the CP.*

*The article proposes the concept of information technology (IT) of building CP based on its representation of PN. The initial information for IT is a verbal description of the algorithm for solving the problem, the output information is a high-quality CP and the corresponding PN.*

*IT includes the following steps: building the original CP and iteratively improving its quality by eliminating common errors and optimizing; decomposition of CP into macro-operations (MO) and compilation of PN fragments for them; modeling of PN fragments; construction of a complete PN and its modeling; saving to the MO library the corresponding fragments of the PN, the debugged complete CP and the corresponding complete PN.*

*The content of the IT stages is described, taking into account the results that have been achieved.*

**Key words:** *software product quality, concept, information technology, stage, verbal description of the algorithm, computational process, decomposition, fragment, Petri net, modeling, optimization, library.*

Статья отправлена: 24.01.2022 г.  
© Комлевая Н.О., Паулин О.Н.