

УДК 004.7

DOI: [10.26102/2310-6018/2019.27.4.020](https://doi.org/10.26102/2310-6018/2019.27.4.020)

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМЫ МНОВОВАРИАНТНОГО СИНТЕЗА ПРОЕКТНЫХ РЕШЕНИЙ ПРИ РАЗРАБОТКЕ АРХИТЕКТУРЫ РАЗВИВАЮЩИХСЯ ПРОГРАММНЫХ СИСТЕМ

А.А.Рындин¹, Н.А.Рындин², С.В.Сапегин³

^{1,2}*Воронежский государственный технический университет, Воронеж,
Российская Федерация*

³*Воронежский государственный университет, Воронеж, Российская Федерация*

^{1,2}*e-mail: alexandr.a.ryndin@me.com*

³*e-mail: sapegin@sc.vsu.ru*

Резюме: В статье рассматриваются модели и алгоритмы синтеза проектных решений в процессе разработки развивающихся программных систем, базирующиеся на подходе многовариантной интеграции. Проблеме эволюции программного обеспечения, как и проблеме проектирования и контроля эволюционной траектории программных систем в настоящее время уделяется большое внимание. Различают различные виды систем, имеющих разные характер и продолжительность эволюционного цикла, однако общей задачей является рациональное проектирование жизненного цикла ПО, включая стадию его разработки. Представив проектируемое ПО, как динамическую развивающуюся систему, мы можем выразить процесс построения программного комплекса в виде маршрута проектирования, состоящего из последовательности задач многокритериального выбора, причем результат каждого выбора влияет на траекторию последующего развития системы. Разделяя компоненты ИС на организационные решения, информационную инфраструктуру и пользовательские сервисы, и применяя методы многовариантной интеграции, мы можем в численном виде решить задачу рационального выбора вариантов на каждом из этапов принятия решений в процессе развития ПО и, таким образом, осуществлять разработку программных средств и компонентов в условиях современной инновационной экономики.

Ключевые слова: многовариантный синтез проектных решений, программная архитектура, развивающиеся программные системы

Для цитирования: Рындин А.А., Рындин Н.А., Сапегин С.В. Математические модели и алгоритмы многовариантного синтеза проектных решений при разработке архитектуры развивающихся программных систем. *Моделирование, оптимизация и информационные технологии*. 2019;7(4):1-10. Доступно по: https://moit.vivt.ru/wp-content/uploads/2019/11/RyndinSoavtors_4_19_1.pdf DOI:10.26102/2310-6018/2019.27.4.020

MATHEMATICAL MODELS AND ALGORITHMS OF MULTIVARIATE SYNTHESIS OF DESIGN SOLUTIONS IN DEVELOPMENT OF ARCHITECTURE OF DEVELOPING SOFTWARE SYSTEMS

A.A.Ryndin¹, N.A.Ryndin², S.V.Sapegin³

^{1,2}*Voronezh State Technical University, Voronezh, Russian Federation*

³*Voronezh State University, Voronezh, Russian Federation*

Abstract: The article discusses models and algorithms of synthesis of design solutions in the process of development of developing software systems, based on the approach of multiple integration. The problem of software evolution, as well as the problem of designing and

controlling the evolutionary trajectory of software systems, is now receiving much attention. Different types of systems have different nature and duration of the evolutionary cycle are distinguished, but the overall objective is to design the software life cycle in a rational manner, including its development stage. Having presented the designed software as a dynamic developing system, we can express the process of building a software complex in the form of a design route consisting of a sequence of tasks of multicriteria selection, and the result of each choice affects the trajectory of subsequent development of the system. By dividing IP components into organizational solutions, information infrastructure and user services, and applying methods of multiple integration, we can solve in numerical form the problem of rational choice of options at each stage of decision-making in the process of software development and thus develop software tools and components in the conditions of modern innovative economy.

Keywords: multimodal data, repository, research organization, distributed platform, interdisciplinary research.

For citation: Ryndin A.A., Ryndin N.A., Sapegin S.V. Mathematical models and algorithms of multivariate synthesis of design solutions in development of architecture of developing software systems. *Modeling, optimization and information technology*.2019;7(4):1-10. Available by: https://moit.vivt.ru/wp-content/uploads/2019/11/RyndinSoavtors_4_19_1.pdf DOI:10.26102/2310-6018/2019.27.4.020 (In Russ.).

Введение

Мир современных информационных технологий весьма разнообразен с точки зрения выполняемых задач, используемых устройств, технологий и языков программирования. Актуальные тенденции цифровизации практически всех без исключения сфер человеческой деятельности требуют развития и совершенствования подходов, направленных на рационализацию разработки программного обеспечения в различных предметных областях. При этом, современные разработки, как правило, не делаются с нуля, а возникают в среде с уже существующими, разработанными ранее библиотеками и компонентами, на основе и с использованием уже работающих решений. Во многих случаях можно говорить о концепции развития единой информационно-сервисной среды в соответствии с эволюционирующими информационными потребностями пользователей. Принимая во внимание итеративный подход к разработке сервисов этой среды, логично будет рассматривать процесс постоянного совершенствования информационно-сервисной среды с точки зрения эволюции ее компонентов под влиянием изменяющихся пользовательских потребностей, внешней среды, технологических и аппаратных требований и т.д.

Проблеме эволюции программного обеспечения в последнее время уделяется все больше и больше внимания в различных источниках. Различают различных виды систем, имеющих разные по времени циклы эволюционного развития (технические системы, платформенные решения, программные системы и т.д.) На Рисунке 1, иллюстрирующем основные направления исследований в современной программной инженерии, показаны перспективы использования эволюционного подхода для изучения аспектов жизненного цикла программных изделий.



Рисунок 1 – Направления исследований современной программной инженерии
Figure 1 – Directions of researches in modern software engineering

Одной из наиболее актуальных задач современной разработки ПО, таким образом, является рационального планирования жизненного цикла, исходя из ресурсных возможностей, а также из нужд бизнеса. При этом, одним из способов управления ЖЦ программного продукта является разработка архитектуры ПО. Поэтому, решение задачи поиска рациональной архитектуры программного средства с учетом его планируемой траектории эволюции является актуальной задачей.

Формализация задачи синтеза вариантов архитектур ИС

В процессе формализации задачи синтеза рациональной архитектуры программного средства с точки зрения его эволюционного развития достаточно быстро можно столкнуться с резким повышением неопределенности в области доступных ресурсов, решаемых задач и организационных моментов. Это бывает особенно актуально для стартапов, чья траектория развития зависит не только от изменения пользовательских требований, но и от особенностей финансирования. Поэтому, изначально ставить задачу формального поиска одного наиболее рационального варианта не представляется эффективным. Необходимость учета системных связей, а также совместного влияния нескольких типов неопределенностей в процессе нахождения рациональных решений приводит к классам моделей, направленных на поиск группы доминирующих вариантов решений, после чего выбор рабочей варианта осуществляется на основе экспертного мнения. Формализация задачи синтеза рациональной архитектуры ИС должна быть осуществлена с учетом этого соображения.

Рассмотрим проектируемый программный комплекс, как динамическую развивающуюся систему. В общем виде класс динамических систем в линейном и конечномерном приближении может быть представлен в виде

$$R(t) = \int_{-\infty}^t K(\tau, t)I(\tau)d\tau \quad (1)$$

где I – вектор входных воздействий, R – вектор реакций системы, K – матрица импульсных переходных функций системы, t – время. В нелинейных случаях K зависит от I . В случае моделирования простейшей РС мы имеем две функции:

1. Внутренняя, обеспечивающая ее существование
2. Внешняя, являющаяся результатом взаимодействия с внешней средой.

Процесс построения программного комплекса, таким образом, можно выразить в виде маршрута проектирования, состоящего из последовательности задач многокритериального выбора, причем результат каждого выбора влияет на траекторию последующего развития системы. Элементами выбора в процессе проектирования корпоративной ИС являются элементы вектора \overline{W}_g , каждый из которых, в свою очередь, содержит в себе набор компонентов, из которых состоит система. Элементы выбора $w_g = \overline{1, W}_g$ в целом задают вариант

$$S_l = (w_1, w_2, \dots, w_g, \dots, w_G) \in S, \quad (2)$$

и характеризуются вектором параметров f_{Wg} . При переходе от одной реализации $w_g \in \overline{1, W}_g$ к другой компоненты вектора f_{Wg} меняются дискретным образом. Оптимальный синтез маршрута проектирования и реализующих его средств путем установления зависимости технико-экономических показателей системы $F_i (i = \overline{1, I})$ от параметров элементов $F_i = \Psi(f_{Wg})$ и определения значений f_{Wg} , обеспечивающих выполнение требований $F_i (i = \overline{1, I})$, в большинстве случаев приводит к параметрическим решениям, в соответствии с которыми не удается поставить определенный маршрут проектирования и реализующую его структуру $w_g \in \overline{1, W}_g$. Поэтому выбор S приходится проводить из множества вариантов ($l = \overline{1, L}$), представляющих собой возможные комбинации конкурирующих технологий и вариантов интеграции $w_g \in \overline{1, W}_g (g \in \overline{1, G})$. При этом, предпочтительность тех или иных комбинаций может меняться за счет различных факторов, что делает задачу выбора варианта в общем случае достаточно плохо структурированной. Рассмотрим один из путей решения сформулированной задачи, основанный на формировании и рационализации модели вариантов программных средств.

Реализация

В составе современных ПС можно выделить следующие подсистемы:

1. Организационный контур, представляющий собой совокупность бизнес-процессов, организационных решений, регламентов и процессов взаимодействия сотрудников предприятия. Организационная подсистема является объектом бизнес-моделирования, автоматизации и реинжиниринга.

2. Корпоративная сеть (КС) – инфраструктура, используемая для манипулирования информацией, которая включает в себя набор аппаратных и программных средств, обеспечивающих информационное взаимодействие пользователей. Является также платформой, на основе которой осуществляются интеграция и дальнейшая эксплуатация функциональных сервисов.

3. Функциональная подсистема, состоящая из взаимосвязанных компонентов, обеспечивающих автоматизацию задач организационного контура. Эта система строится на базе КС и привносит в систему прикладную функциональность, представляющую для пользователя главную ценность. Требования к ней зачастую сложны и противоречивы, однако в конечном счете именно использование этой составляющей является способом

повышения эффективности функционирования организации путем решения конкретных задач автоматизации тех или иных бизнес-процессов предприятия.

Компоненты этих трех подсистем на практике достаточно сильно связаны как внутри подсистемы, так и с компонентами других подсистем. Более того, характер причинно-следственных связей, возникающих между компонентами, является достаточно сложным. Так, набор организационных процессов определяется возможностями инфраструктурной и функциональной подсистем. В свою очередь инфраструктура ИС может как определяться требованиями со стороны функциональной подсистемы, так и сама определять границы допустимых вариаций функциональных подсистем. Состав функциональной подсистемы может как определять требования к инфраструктуре ИС, так и зависеть от этих требований, в зависимости от соотношения затрат на ввод в эксплуатацию компонентов подсистем и политики предприятия в области ИТ. При построении эффективной корпоративной ИС необходим учет взаимного влияния во всех вышеперечисленных случаях. Формализацию модели можно осуществлять, исходя из представления на Рисунке 2. При этом, модель состоит из следующих подсистем:

1. Организационные структуры процессов $a_i, i \in (1, N)$, где N – общее количество возможных организационных структур;
2. Элементы сетевой и платформенной инфраструктуры $b_j, j \in (1, M)$, где M – общее количество платформенных подсистем, которые можно использовать в работе корпоративной ИС;
3. Функциональные компоненты (программные сервисы) корпоративной ИС $c_k, k \in (1, L)$, где L – общее количество программных компонентов, которые возможно использовать на каждом этапе развития системы. Каждая конфигурация корпоративной ИС может состоять из произвольного количества подсистем a_i, b_j и c_k .

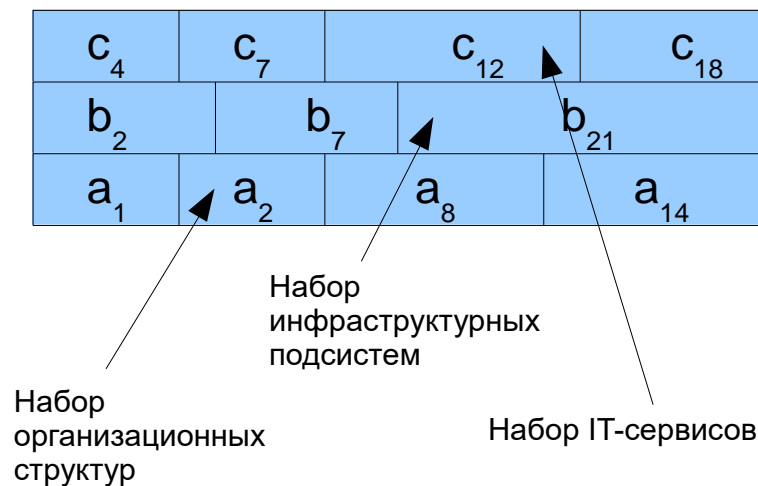


Рисунок 2 - Концептуальная модель структуры ИС
 Figure2 – Concept model of Information System structure

Базируясь на этом представлении, саму структуру ИС можно представить в виде ориентированного мультиграфа, в котором вершинами являются компоненты a_i, b_j и c_k , а дуги олицетворяют зависимости между этими компонентами. В этом случае модель архитектуры ИС представляет собой дискретно-детерминированный автомат, который в каждый момент времени T осуществляет переход в новое состояние, характеризуемое

составом компонентов и набором связей между ними. Постановка задачи рационального развития структуры ИС подразумевает собой задачу выбора на каждом этапе таких состояний из множества возможных, прохождение которых повышает качество структуры системы. При этом, проектировщик системы имеет возможность влиять на состав компонентов c_k и частично - на некоторые типы зависимостей и инфраструктурные компоненты b_j .

Численные методы решения задачи

Рассмотрим подробнее процесс формирования матрицы условных вероятностей предпочтительного использования множеств вариантов интеграции на каждом из уровней и между уровнями. В условиях неопределенности структуры отдельных компонентов системы, уникальности организационных процессов и разрабатываемых ПС определение матрицы условных вероятностей использования во многих случаях невозможно из-за недостатка информации. Использование в этих случаях экспертного опыта, как показывает практика, не способно решить проблему в целом. Поэтому, предпочтительность совместного использования предлагается вычислять не на основе статистики, а на основе использования метрических показателей, определяющих структурные, процессные, объектно-ориентированные и прочие особенности программных компонентов. Для множества организационных процессов в общем виде необходимо также осуществить классификацию, исходя из принятых в IT-индустрии метрик, оценивающих процессы. Подобный подход позволяет не только преодолеть во многих случаях критичный недостаток данных, но и обеспечить источник экспертной информации для последующего сравнения вариантов.

Определим для подсистем организационных структур, инфраструктурных и программных компонентов соответственно наборы метрик, с помощью которых можно осуществить классификацию. Для компонентов ПО можно выделить следующие группы метрик:

1. Метрики сложности потока команд (цикломатическая метрика Маккейба, метрика Майерса, метод Хансена, метрика Пивоварского, метрика Шнейдевинда, топологическая мера Чена, мера Вудворта).
2. Метрики сложности потока данных (метрика Чепина, информационная мера сложности относительно структуры данных, мера Овието, метрика спена).
3. Структурные метрики (тестирующая мера, мера связанности модулей программы по данным, по структуре, по управлению, по содержимому, по внешним данным; метрика Мак-Клура, мера Берлингера).
4. Объектно-ориентированные метрики (метрики Мартина, метрики Чидамбера и Кемерера).
5. Гибридные метрики (метрика Кокола, метрики Зольновского, Симмонса, Тейера).

Конкретный набор метрик для решения задач рационального проектирования программных систем в различных предметных областях и различных типах архитектур может быть составлен, исходя из следующих принципов:

1. Вычислимость. Наиболее предпочтительно использовать метрики, вычисление которых в рассматриваемых условиях является более простым процессом.

2. Сбалансированность. Необходимо использовать, по крайней мере, одну метрику каждой группы для достижения сбалансированной картины оценки программных компонентов.

3. Экспертный опыт.

Для оценки подсистем организационных структур, нацеленных на обеспечение конкретных производственных процессов, а также для оценки уровня сервисного обеспечения, предоставляемого платформенным ПО, целесообразно использовать следующие метрики организационных процессов:

- ресурсоемкость процесса (кол-во требуемых ресурсов);
- важность процесса (экспертная оценка);
- степень удовлетворенности клиентов (экспертная оценка);
- количество инцидентов (на единицу времени);
- число запросов на изменение;
- скорость изменения процессов (кол-во изменений на единицу времени);
- число уникальных решений (число решений задачи в обход процесса);
- уровень сотрудников (кол-во дней на обучение);
- количество документов, поддерживающих процесс (количество документов, не используемых повторно).

Оценка функционального контура, в случае использования сервисного подхода на основе единой платформы может выполняться с помощью следующих метрик:

1. % расходов на поддержание ИТ-структуры в общей стоимости решения;
2. % фактической загрузки ИТ-ресурсов в нормальном режиме работы;
3. Время простоя системы (продолжительность в единицу времени);
4. Среднее время до наступления инцидента;
5. Накладные расходы
6. Количество нарушений SLA (на единицу времени).

Исходя из определенных наборов метрик, для базовых наборов компонентов платформенного и функционального решения (а также, отчасти, и для организационных решений) можно сформировать следующую схему исследования:

1. Формирование множеств компонентов А,В,С.
 2. Формирование на основе сочетаний компонентов множеств вариантов $E=(A_n, B_m, C_k)$. Отсевание заведомо неправдоподобных вариантов.
 3. Формирование массива оценок существующих вариантов (E, Mx) на основе определенных наборов метрик;
 3. Определение целевых метрик и ранжирование диапазонов их значений (по уровню предпочтительности характеристик для вариантов систем);
 4. Построение отношения предпочтения на множестве вариантов E. Нормирование этого отношения.
 5. Формирование матрицы совместного использования компонентов на основе множества вариантов E (считая все существующие варианты совместного использования равновероятными).
 6. Модификация матрицы совместного использования на основе построенного отношения предпочтения.
 7. На основе готовой матрицы совместного использования рассчитывается матрица совместной энтропии компонентов А,В,С.
- Соответственно, задача поиска рациональных вариантов архитектуры программного решения сводится к решению задачи ограничения разнообразия вариантов организации платформы ИС \bar{B} и слоя программных сервисов \bar{C} для заданного множества вариантов организации процессов $\bar{A} = A_l$.

Заключение

Разработка архитектур современных развивающихся программных систем является сложной, плохо структурированной задачей, для решения которой необходимо принимать во внимание целый ряд различных факторов. Предлагаемый подход к синтезу рациональной структуры ПС на основе трехуровневой модели архитектуры корпоративных ИС, основанной на принципах сервисного подхода к описанию составляющих ее подсистем, безусловно, не является единственным, но позволяет с приемлемыми трудозатратами определить основные характеристики проектируемой ИС на количественном уровне. Матрица взаимной энтропии структурных элементов может быть рассчитана на основе использования наборов метрик, характеризующих свойства тех или иных компонентов, а также из совместного использования. Таким образом, на основе общей модели архитектуры ИС разработан метод решения задачи построения рациональной программной архитектуры, отличающийся использованием многовариантной интеграции и позволяющий осуществлять выбор рационального варианта архитектуры программного комплекса

Значимость корректно построенной архитектуры в рамках проекта разработки ПО достаточно велика, однако не меньшее значение для успеха представляет собой рациональный выбор парадигмы разработки программного средства, включающей в себя бэкграунд в виде используемых технологий и средств разработки, шаблонов и подходов к решению задач проектирования в каждом конкретном случае.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта 19-07-01251, конкурс А (договор от 16.01.2019 г.)

ЛИТЕРАТУРА

1. Altshuller Genrich (2005). 40 Principles: extended edition. Translated by Lev Shulyak with additions by Dana Clarke. Sr. Worcester, MA: Technical Innovation Center. ISBN 978-0-9640740-5-7
2. Mario Hermann, Tobias Pentek, Boris Otto. Design Principles for Industrie 4.0 Scenarios. 49th Hawaii International Conference on System Sciences (HICSS). 2016.
3. Klaus Schwab. The Fourth Industrial Revolution. World Economic Forum. 2016. ISBN 1944835008.
4. Klaus Schwab. The Fourth Industrial Revolution: what it means, how to respond. World Economic Forum. 2017.
5. Львович Я.Е. Многоальтернативная оптимизация: теория и приложения. Воронеж: Кварта; 2006.
6. Методология Rapid Foresight 0.4, АСИ, КСП. Москва, 2017
7. Рындин А.А. Многовариантная интеграция: теория и приложения в САПР. Воронеж, ВГТУ; 2018.
8. Object Management Group, Essence - Kernel And Language For Software Engineering Methods (Essence). 2014. Available by: <http://www.omg.org/spec/Essence/>.
9. Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence and Svante Lidman. The Essence of Software Engineering: The SEMAT Kernel. Communications of the ACM. 2012;55(12):42-49.

REFERENCES

1. Altshuller Genrich (2005). 40 Principles: extended edition. Translated by Lev Shulyak with additions by Dana Clarke. Sr. Worcester, MA: Technical Innovation Center. ISBN 978-0-9640740-5-7
2. Mario Hermann, Tobias Pentek, Boris Otto. Design Principles for Industrie 4.0 Scenarios. 49th Hawaii International Conference on System Sciences (HICSS). 2016.
3. Klaus Schwab. The Fourth Industrial Revolution. World Economic Forum. 2016. ISBN 1944835008.
4. Klaus Schwab. The Fourth Industrial Revolution: what it means, how to respond. World Economic Forum. 2017.
5. Lvovich Ya.E. Multi-alternative optimization: theory and application. Voronezh: Kvarta, 2006.
6. Rapid Foresight Methodology 0.4 COMMUNITIES OF PRACTICE DEVELOPERS. Moscow, 2017.
7. Ryndin A.A. Multiple integration: theory and applications in CAD: monograph. Voronezh: Voronezh State Technical University, 2018.
8. Object Management Group, Essence - Kernel And Language For Software Engineering Methods (Essence). 2014. Available by: <http://www.omg.org/spec/Essence/>.
9. Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence and Svante Lidman. The Essence of Software Engineering: The SEMAT Kernel. Communications of the ACM. 2012;55(12):42-49.

ИНФОРМАЦИЯ ОБ АВТОРЕ / INFORMATIONS ABOUT AUTHORS

Рындин Александр Алексеевич, доктор техн. наук, профессор, ФБГОУ ВПО Воронежский государственный технический университет, г. Воронеж, Российская Федерация.

Alexander A. Ryndin, PhD, Doctor in Technical Sciences, Voronezh State Technical University, Voronezh, Russian Federation.

Рындин Никита Александрович, канд. техн. наук, доцент, ФБГОУ ВПО Воронежский государственный технический университет, г. Воронеж, Российская Федерация.

Nikita A. Ryndin, PhD in Technical Sciences, Voronezh State Technical University, Voronezh, Russian Federation.

Сапегин Сергей Владимирович, канд. техн. наук, ФБГОУ ВПО Воронежский государственный университет, г. Воронеж, Российская Федерация.

Sergey V. Sapegin, PhD in Technical Sciences, Voronezh State University, Voronezh, Russian Federation.