

Повышение эффективности разработки распределенных систем автоматизированного проектирования с применением диакоптики*

В. Н. Гридин^I, В.И. Анисимов^{I,II}, С.А. Васильев^{I,II}

^I ФГБУН Центр информационных технологий в проектировании РАН, г. Одинцово, Московская обл., Россия

^{II} Санкт-Петербургский электротехнический университет, г. Санкт-Петербург, Россия

Аннотация. Описаны основные стадии жизненного цикла информационных систем, в частности распределенных систем автоматизированного проектирования электронных схем. Представлены современные модели жизненного цикла, преимущества и сценарии их использования. Определены инструменты автоматизации с примерами каждой основной стадии жизненного цикла информационных систем. Предложена методика моделирования слабосвязанных электронных схем на основе технологии расчета больших электронных схем по частям. Показано, что реализация диакоптического подхода к моделированию больших слабосвязанных схем существенно увеличивает производительность вычислительного процесса, что особенно актуально при разработке распределенных систем автоматизированного проектирования. Предложено для построения и преобразования описания моделируемой схемы использовать обобщенные сигнальные графы, отображающие уравнения схемы в обобщенной причинно-следственной форме.

Ключевые слова: жизненный цикл информационных систем, инструменты автоматизации разработки информационных систем, DevOps, системы автоматизированного проектирования, моделирование схем, обобщенный сигнальный граф, методы диакоптики.

DOI 10.14357/20718632220101

Введение

Применение микросервисной архитектуры в разработке программного обеспечения информационных систем, и в частном случае распределенных систем автоматизированного проектирования электронных схем, позволяет обеспечить высокую эффективность и отказоустойчивость, однако для их построения не существует единого инструмента. Стоит отметить, что реализация данной концепции возможна с применением различных технологий, как сочетающихся, так

и параллельных. Актуальной задачей для команд разработки распределенных систем на основе микросервисной архитектуры является определение средств автоматизации для каждого основного этапа жизненного цикла, охватывающих наибольший спектр применяемых технологий с целью повышения эффективности разработки и развертывания программного продукта. Наиболее эффективным способом повышения производительности программного обеспечения схемотехнических САПР для моделирования больших схем является разделение

* Работа выполняется в рамках темы № FFSM-2019-0001.

моделируемой схемы на составляющие под-схемы с последующим формированием и преобразованием для каждой из подсхем необходимого математического описания. Общее решение для всей схемы получается путем соединения полученных частных решений для подсхем [1-3]. При этом не требуется выполнять формирование полной системы уравнений для всей схемы, достаточно сформировать и соответствующим образом обработать уравнения для ее элементарных подсхем, которые могут быть сделаны настолько малыми, насколько это требуется для расчета конкретной схемы. Использование диакоптического подхода для расчета больших слабосвязанных электронных схем является особенно актуальным при разработке распределенных систем автоматизированного проектирования, поскольку при работе с сети Интернет необходимо обеспечить достаточно высокое быстродействие распределенной системы, чтобы минимизировать время взаимодействия пользователя с сетью [4-7].

1. Автоматизация разработки информационных систем и систем проектирования

Жизненный цикл программного обеспечения, согласно российским и международным стандартам, является периодом времени от принятия решения о создании продукта до момента полного изъятия его из эксплуатации [8, 9]. Постановка цели и планирование работ в любой деятельности, как и в сфере построения программных систем, позволяет повысить эффективность процессов и снизить количество затраченных ресурсов. Существуют шесть основных стадий жизненного цикла программного обеспечения: анализ, составление требований к продукту; планирование; проектирование и дизайн; разработка; тестирование; развертывание, эксплуатация [10]. В связи с различными факторами (технологическими, экономическими, временными) проекта программного продукта, используются различные модели жизненного цикла, определяющие взаимосвязь его стадий. Ниже представлены основные модели жизненного цикла:

1. Каскадная модель — последовательный поток, проходящий стадии жизненного цикла программного обеспечения. Эта модель, как правило, используется при построении относительно простых программных систем, так как не обеспечивает возможности возврата к предыдущим шагам.

2. V-образная модель — последовательный поток, проходящий стадии со строгим контролем качества на каждой из стадий.

3. Модель эволюционного прототипирования — последовательное выполнение стадий с возвратом на любую предшествующую с целью корректировки и создания нового прототипа программного продукта.

4. Инкрементная модель — линейная последовательность выполнения стадий для каждого запланированного улучшения продукта. Модель обеспечивает инкремент новой функции в продукт.

5. Спиральная модель — последовательное выполнение стадий в рамках витка спирали для создания очередной версии продукта, уточняются требования проекта и осуществляется новый виток, производится новая версия продукта (Рис. 1).

Применение методологии автоматизации технологических процессов (стадий) жизненного цикла программного обеспечения позволяет существенно увеличить скорости разработки и публикации для потребителя, а также повысить степень отказоустойчивости программных систем. Основной задачей данного подхода под названием DevOps является интеграция всех стадий жизненного цикла в единый автоматизированный процесс. Ввиду того, что среды разработки и эксплуатации весьма многочисленны и разнообразны, единого инструмента автоматизации не существует, пути определения таких инструментов достаточно индивидуальны. Отсутствие готовых шаблонов и индивидуальность настройки различных систем делают специалистов в области автоматизации технологических процессов жизненного цикла программного обеспечения весьма востребованными специалистами. Отличительной особенностью специалистов в области DevOps является наличие системного мышления, то есть понимание того, как определенные действия

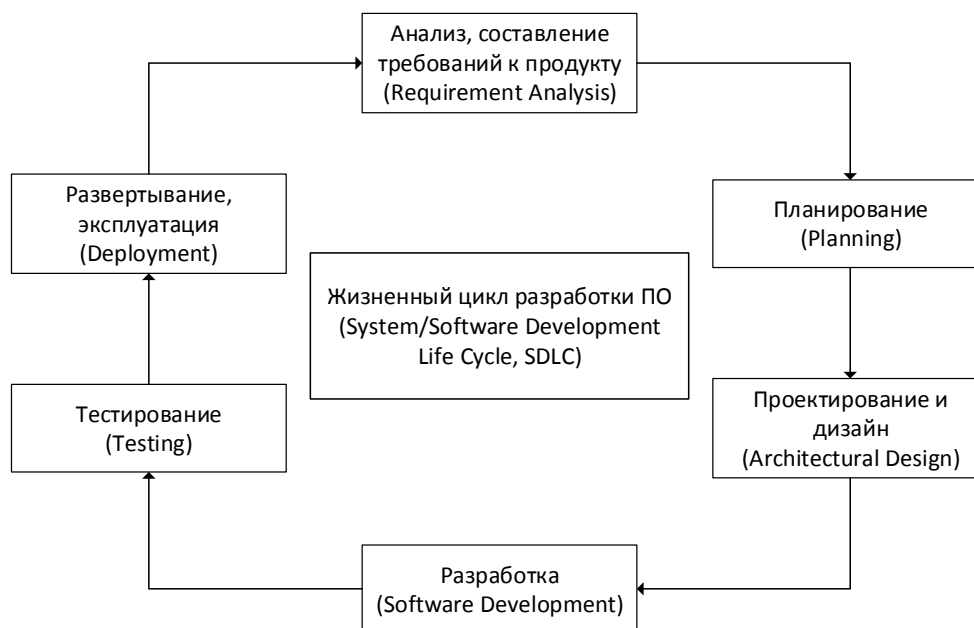


Рис. 1. Основные этапы жизненного цикла разработки ПО (SDLC)

вливают на всю систему в целом в отношении бизнеса, инженерии и менеджмента на макро и микроуровнях. Однако стоит отметить, что некоторые наборы инструментов по автоматизации и контролю процессов жизненного цикла для построения распределённых веб-ориентированных информационных систем все же существуют.

Ниже приводится описание некоторых наборов инструментов автоматизации разработки и контроля процессов, соответствующих основным стадиям жизненного цикла программного обеспечения.

Инструменты автоматизации разработки. К ним относятся программное обеспечение по управлению исходным кодом (SCM), системы непрерывной интеграции (CI), программное обеспечение для осуществления контроля версий и управления миграциями БД. Наиболее распространёнными типами SCM являются продукты на основе распределённой системы управления версиями GIT, например программный продукт GitLab. Отличительной особенностью ПО GitLab является возможность установки на собственные серверы дистрибутива по подписке (On-premise). Важной особенностью программного обеспечения CI при построении информационных систем на основе

микросервисной архитектуры является поддержка работы в контейнерной среде, с чем может справиться программный продукт GitLab CI. Программное обеспечение для осуществления контроля версий и управления миграциями БД обеспечивает возможность гибкой работы по модернизации схемы БД и поддержке клиентских приложений.

Инструменты автоматизации тестирования. Данный инструмент основывается на методологии разработки через поведение (Behavior-driven development, BDD), это программное обеспечение использует предметно-ориентированный язык для упрощения обмена информацией между группами бизнес-задач и технических задач. Эта методология базируется на подходе разработки через тестирование (test-driven development, TDD) с введением следующего порядка: для создания любого теста требуется описать необходимый результат новой функции на предметно-ориентированном языке. Для описанных задач может подойти программное обеспечение Cucumber, адаптированное для различных языков программирования и предоставляющее описанный функционал.

Инструменты автоматизации развертывания. К ним можно отнести программное обеспечение по управлению артефактами, системы

управления конфигурацией, системы автоматизации развертывания. Артефактами являются вспомогательные элементы продукта, входящие в его состав. Например, скомпилированное приложение, библиотеки или документация, как правило, все то, что имеет двоичное, а не текстовое состояние и по этой причине не подходящее для использования в системах управления исходным кодом, но также требующее контроля версий для обеспечения работы программного продукта. Удачным решением для управления артефактами может стать программное обеспечение Nexus, этот инструмент поддерживает большое количество технологий разработки информационных систем. К этим технологиям можно отнести менеджер пакетов NPM с возможностью автоматизированного управления и контейнеризатор Docker. Системы управления конфигурацией отвечают за выделение облачных ресурсов и оркестровку внутри службы. К таким декларативным системам относится программное обеспечение Ansible, в котором для описания заданий автоматизации используется формат на основе языка YAML. Непосредственно к системам автоматизации развертывания можно отнести программное обеспечение, производящее балансировку и управление ресурсами серверов, таблицами маршрутизации, записями доменных имен DNS. К таким оркестраторам с открытым исходным кодом можно отнести мощнейший инструмент Kubernetes для работы с контейнерными микросервисными системами, или инструмент Terraform для различных типов систем.

Инструменты среды выполнения. Важнейшей задачей любого проекта разработки программного обеспечения является запуск прило-

жения в производство, но для его работы требуется различная инфраструктура, которую обеспечивают соответствующие провайдеры. Не так давно появился новый термин «все, как услуга» XaaS, описывающий полный спектр предоставляемых ИТ услуг, таких как инфраструктура, платформа, софт, мониторинг, коммуникации и даже кибератаки, представляющиеся как сервис (MaaS). Преимущества масштабируемости, времени запуска проекта и производительности очевидны, однако существуют и недостатки – стоимость, конфиденциальность, непрогнозируемая отказоустойчивость.

Описанные наборы инструментов позволяют автоматизировать процессы основных стадий жизненного цикла программного обеспечения и оптимизируют технологические, экономические и временные расходы программного продукта.

2. Математическое описание больших электронных схем на основе диакоптики

В моделируемой схеме всегда можно провести декомпозицию ее структуры путем выделения слабосвязанных подсхем, между которыми отсутствуют индуктивные связи и выполняется условие концентрации управляющих и управляемых переменных зависимых источников в пределах отдельной подсхемы (Рис. 2).

Для каждой выделенной k -й подсхемы можно ввести внутренние переменные x_i и образовать для каждой подсхемы вектор внутренних переменных $X_k = [\dots, x_i, \dots]^T$. Для каждого узла схемы можно ввести узловую переменную v_p , определяющую состояние этого узла, и из всех введенных узловых переменных образовать вектор

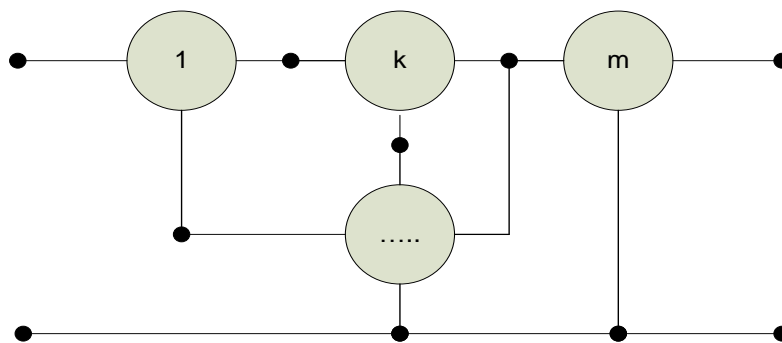


Рис. 2. Декомпозиция слабосвязанной схемы

узловых потенциалов $X_0 = V_0 = [\dots, v_p, \dots]^T$. Если провести нумерацию переменных таким образом, что в начале нумеруются переменные отдельных подсхем, а на заключительном этапе узловые переменные связи между подсхемами, то уравнения моделируемой схемы будут содержать матрицу с окаймленной блочно-диагональной структурой:

$$\begin{array}{|c|c|c|c|c|c|} \hline W_{11} & & & & & W_{10} \\ \hline & \dots & & & & \dots \\ \hline & & W_{kk} & & & W_{k0} \\ \hline & & & \dots & & \dots \\ \hline & & & & W_{mm} & W_{m0} \\ \hline W_{0l} & \dots & W_{0k} & \dots & W_{0m} & W_{00} \\ \hline \end{array} \begin{array}{|c|} \hline X_1 \\ \hline \dots \\ \hline X_k \\ \hline \dots \\ \hline X_m \\ \hline X_0 \\ \hline \end{array} + \begin{array}{|c|} \hline S_1 \\ \hline \dots \\ \hline S_k \\ \hline \dots \\ \hline S_m \\ \hline S_0 \\ \hline \end{array} = 0 \quad (1)$$

Система уравнений для k -й подсхемы имеет вид:

$$W_{kk} X_k + W_{k0} X_0 + S_k = 0, \quad k = \overline{1, m}. \quad (2)$$

Отсюда можно получить выражения для вектора внутренних переменных X_k отдельных подсхем:

$$X_k = -W_{kk}^{-1} W_{k0} X_0 - W_{kk}^{-1} S_k, \quad k = \overline{1, m}.$$

Подставляя это выражение в последнюю блочную строку окаймленной матрицы в уравнении (1), получим уравнение для определения вектора переменных узлов связи:

$$\bar{W}_{00} X_0 + \bar{S} = 0,$$

где $\bar{W}_{00} = W_{00} - \sum_{k=1}^m W_{0k} \bar{W}_{k0}, \quad \bar{S}_0 = S_0 - \sum_{k=1}^m W_{0k} \bar{S}_k,$

$$\bar{W}_{k0} = W_{kk}^{-1} W_{k0}, \quad \bar{S}_k = W_{kk}^{-1} S_k.$$

Таким образом, вместо решения общего уравнения схемы $WX + S = 0$, что требует вычисления обратной матрицы для общей матрицы схемы W , необходимо выполнить обращение матриц W_{kk} и W_{00} , порядок которых может быть значительно снижен путем выбора при декомпозиции исходной схемы достаточно большого числа подсхем. При этом нет необходимости в составлении полной системы уравнений, требуется только сформировать уравнения для ее подсхем и для переменных узлов связи.

Однако отмеченные преимущества диакоптического подхода к расчету больших слабо-связанных схем будут сохраняться только при

условии неизменности сохранения описания моделируемой схемы в виде блочно-диагональной окаймленной структуры (1). Вместе с тем, в реальных условиях в процессе проектирования схемы неизбежно будут появляться новые переменные, что приведет к нарушению первоначальной блочно-диагональной структуры и потребует полной перекодировки переменных, чтобы сохранить окаймленную структуру описания схемы.

Для решения указанной проблемы целесообразно использовать топологическое описание моделируемой схемы при помощи обобщенных сигнальных графов [11], которые отображают систему исходных уравнений в обобщенной причинно-следственной форме и содержат взвешенные вершины. Помимо наглядного представления о структуре связей между подсхемами описание задачи моделирования на основе обобщенных сигнальных графов позволяет легко решить проблему выбора последовательности нумерации внутренних переменных подсхем и переменных узлов связи моделируемой схемы.

В обобщенной причинно-следственной форме уравнение для k -й подсхемы может быть записано в виде:

$$W_{kk} X_k = -W_{k0} X_0 - C_k S_k = 0, \quad k = \overline{1, m}, \quad (3)$$

где W_{kk}, W_{k0} - матрицы параметров k -й подсхемы, S_k - вектор задающих источников k -й подсхемы, C_k - матрица соединений задающих источников k -й подсхемы.

Аналогично, для узлов связи между подсхемами уравнение в обобщенной причинно-следственной форме имеет вид:

$$W_{00} X_0 = -\sum_{k=1}^m W_{0k} X_k - C_0 S_0, \quad (4)$$

где W_{00}, W_{0k} - матрицы параметров узлов связи между подсхемами, S_0 - вектор задающих источников, C_0 - матрица соединений задающих источников, подключенных к узлам связи.

Уравнениям (3) и (4) соответствует приведенный на Рис. 3 обобщенный сигнальный граф.

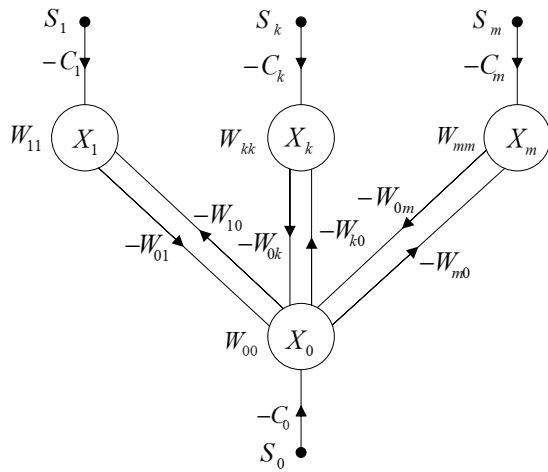


Рис. 3. Обобщенный сигнальный граф моделируемой схемы

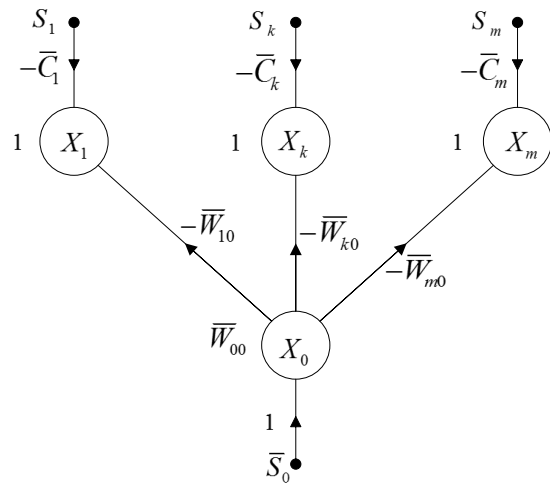


Рис. 4. Заключительное представление топологического описания схемы

Построенный обобщенный сигнальный граф отображает взаимодействие между отдельными подсхемами и переменными связи моделируемой схемы и позволяет обеспечить независимую нумерацию переменных и обработку описания отдельных подсхем, а также подключение новых подсхем к узлам связи.

Умножая уравнение (2) на обратную матрицу W_{kk}^{-1} , можно привести это уравнение к виду:

$$X_k = -\bar{W}_{k0} X_0 - \bar{C}_k S_k, \quad k = \overline{1, m}, \quad (5)$$

где $\bar{W}_{k0} = W_{kk}^{-1} W_{k0}$, $\bar{C}_k = W_{kk}^{-1} C_k$

Для построения заключительного топологического описания задачи подставим уравнение (5) в уравнение (4), что дает:

$$\bar{W}_{00} X_0 = \bar{S}_0, \quad (6)$$

где $\bar{W}_{00} = W_{00} - \sum_{k=1}^m W_{0k} \bar{W}_{k0}$,

$$\bar{S}_0 = -C_0 S_0 + \sum_{k=1}^m W_{0k} \bar{C}_k S_k,$$

Уравнениям (5) и (6) соответствует обобщенный сигнальный граф, приведенный на Рис. 4.

По виду этого слабосвязанного графа можно легко найти значения переменных вектора связей X_0 и векторов X_k внутренних переменных подсхем.

Практическая реализация построения заключительного топологического описания возможна непосредственно по виду исходного обобщенного сигнального графа моделируемой схемы путем топологического исключения из него всех ветвей, исходящих из подграфов, содержащих подмножество вершин X_k подсхем, и нормирования значений весовых функций обобщенных взвешенных вершин $W_{kk} = 1$. При выполнении указанных операций будут пересчитываться весовые функции всех ветвей, входящих в обобщенные вершины X_k .

Следует иметь в виду, что построение заключительной структуры топологического описания схемы возможно только при выполнении условия существования для всех подсхем обратной матрицы W_{kk}^{-1} . Очевидно, что для этого необходимо выполнение условия неравенства нулю определителя Δ матриц W_{kk} для всех подсхем. Эти условия обычно выполняются, если при декомпозиции исходной схемы в качестве отдельных подсхем выделять законченные функциональные блоки, каждый из которых выполняет вполне определенную задачу. В исключительных случаях может потребоваться выполнение отдельного предварительного анализа устойчивости функционирования отдельных подсхем [12, 13].

Топологическое представление описания моделируемой схемы в виде обобщенного сигнального графа предоставляет широкие воз-

возможности для анализа топологической структуры графа и всех его элементарных графов g_i . Каждый элементарный граф содержит совокупность некасающихся контуров и взвешенных вершин (включая взвешенные вершины с нулевым весом). Определитель обобщенного сигнального графа Δ можно вычислить по топологической формуле $\Delta = \sum \delta_i$, где δ_i - определитель элементарного графа, равный произведению весов входящих в него некасающихся контуров и взвешенных вершин [11].

Эффективность моделирования больших схем на основе диакоптического подхода можно оценить коэффициентом экономии оперативной памяти и степенью повышения быстродействия вычислительных операций.

Пусть V_D - объем памяти, которая требуется для расчета схемы с использованием методов диакоптики, V - объем памяти, требуемый для расчета этой схемы обычными методами без выполнения декомпозиции моделируемой структуры. Если схема разделена на m подсхем, число переменных в каждой подсхеме равно n_k , а число переменных связи составляет n_0 , то коэффициент экономии оперативной памяти $\alpha = V/V_D$ при диакоптическом подходе можно определить выражением:

$$\alpha = \frac{(n_0 + \sum_{k=1}^m n_k)^2}{\max_k n_k^2 + 2n_0^2 \sum_{k=1}^m n_k + n_0^2}.$$

Если схема разделена на 10 одинаковых подсхем ($m=10$) и число переменных в каждой подсхеме равно числу переменных связи $n_k = n_0$, то получим значение коэффициента экономии оперативной памяти $\alpha=6$.

Для оценки степени повышения быстродействия на основе диакоптического подхода следует иметь в виду, что время решения уравнений определяется числом мультипликативных операций и пропорционально величине $n^3/3$ для метода Гаусса и $n^3/2$ для метода Гаусса-Жордана. Следовательно, эффективность по

быстродействию может быть определена выражением:

$$\beta = \frac{(n_0 + \sum_{k=1}^m n_k)^3}{\sum_{k=1}^m n_k^3 + bn_0^3 \sum_{k=1}^m n_k + n_0^3},$$

где $b=1/3$ или $b=1/2$ в зависимости от выбранного метода решения уравнений.

Если моделируемая схема разбита на 10 одинаковых подсхем и число переменных в каждой подсхеме равно числу переменных связи, т.е. $n_k=n_0$, то значение степени повышения быстродействия при решении задачи методами диакоптики $\beta = 30$.

Таким образом, диакоптический подход к решению задачи моделирования больших слабосвязанных схем обеспечивает существенное увеличение коэффициента экономии оперативной памяти и значительно повышает быстродействие при решении задач моделирования.

Заключение

Описанные инструменты автоматизации основных стадий жизненного цикла информационных систем и, в частности, систем схемотехнического проектирования, позволяют внедрить методологию разработки программного обеспечения DevOps, позволяя существенно повысить эффективность разработки и развертывания программных продуктов.

Показано, что диакоптический подход к моделированию больших слабосвязанных схем на основе декомпозиции исходной схемы на ряд составляющих подсхем приводит к существенному повышению производительности системы автоматизации схемотехнического проектирования. Использование методов диакоптики обеспечивает значительную экономию затрат оперативной памяти и существенно снижает время решения задач моделирования больших слабосвязанных схем.

Литература

1. Г. Крон Исследование сложных схем по частям – диакоптика, пер с англ. Издательство «Наука», Главная редакция физико-математической литературы. М.: 1972. 544 с.

2. Х. Хэпп Диакоптика и электрические цепи, пер. с англ. -М.: Изд. «Мир» 1974. 342 с.
3. Б.В. Баталов, Ю.Б. Егоров, С.Г. Русаков Основы математического моделирования больших интегральных схем на ЭВМ. М.: Радио и связь, 1982. 168 с.
4. В.И. Анисимов, В.Н. Гридин Методы построения схем автоматизированного проектирования на основе Интернет-технологий и компактной обработки разреженных матриц // Информационные технологии в проектировании и производстве. 2009. №1. С. 3-7.
5. О.С. Коваленко, В.М. Курейчик Обзор проблем и состояний облачных вычислений и сервисов // Известия ЮФУ. Технические науки. 2012. № 7. С. 146-153.
6. В.Н. Гридин, Г.Д. Дмитриевич, Д.А. Анисимов Построение схем автоматизированного проектирования на основе Web-технологий // Информационные технологии. 2011. №5. С. 23-27.
7. Д.А. Анисимов Методы построения схем автоматизации схемотехнического проектирования на основе веб-сервисов // Известия СПбГЭТУ «ЛЭТИ» 2012. №10. С. 56-61.
8. ГОСТ Р 34.601-90. ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ. Комплекс стандартов на автоматизированные системы // Автоматизированные системы стадии создания. М.: Стандартинформ, 2009.
9. ISO/IEC 12207:2008. Systems and software engineering — Software life cycle processes
10. James Taylor Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware, and Integration Initiatives. Delhi: PHI Learning, 2004.
11. В.И. Анисимов Топологический расчет электронных схем. Л.: «Энергия», 1977. 238 с.
12. В.Н. Гридин, В.Б. Михайлов, Л.Б. Шустерман Численно-аналитическое моделирование радиоэлектронных схем // Наука, 2008. 339 с.
13. Б.А. Калабеков, И.Ю. Лapidус, В.М. Малафеев Методы автоматизированного расчета электронных схем в технике связи. М.: «Радио и связь», 1990. 272 с.

Гридин Владимир Николаевич Центр информационных технологий в проектировании РАН, научный руководитель, д.т.н., профессор. Количество печатных работ: 450 (в т.ч. 9 монографий). Область научных интересов: системы автоматизированного проектирования. E-mail: info@ditc.ras.ru,

Анисимов Владимир Иванович Центр информационных технологий в проектировании РАН, главный научный сотрудник, д.т.н., профессор. Санкт-Петербургский Государственный Электротехнический университет, д.т.н., профессор. Количество печатных работ: 400 (в т.ч. 40 монографий). Область научных интересов: системы автоматизированного проектирования. E-mail: info@ditc.ras.ru

Васильев Сергей Алексеевич Центр информационных технологий в проектировании РАН, старший научный сотрудник, к.т.н.. Санкт-Петербургский Электротехнический университет, ассистент, к.т.н. Количество печатных работ: 28. Область научных интересов: информационные системы. E-mail: info@ditc.ras.ru

Increasing the Efficiency Development of Distributed Computer-Aided Design Systems Using Diacoptics

V. N. Gridin¹, V. I. Anisimov^{1,II}, S. A. Vasiliev^{1,II}

¹ Design Information Technologies Center Russian Academy of Sciences, Odintsovo

^{II} Saint Petersburg electrotechnical University, Saint Petersburg

Abstract. The main stages of the life cycle of information systems are described, in particular, distributed systems for computer-aided design of electronic circuits. Modern life cycle models are presented: advantages and scenarios for their use. Automation tools are defined with examples of each main stage of the life cycle of information systems. A technique for modeling weakly coupled electronic circuits based on the technology of calculating large electronic circuits in parts is proposed. It is shown that the implementation of the diacoptic approach to modeling large loosely coupled circuits significantly increases the productivity of the computational process, which is especially important in the development of distributed computer-aided design systems. It is proposed to build and transform the description of the simulated circuit to use generalized signal graphs that display the equations of the circuit in a generalized cause-and-effect form.

Keywords. Information systems life cycle, information systems development automation tools, DevOps, computer-aided design systems, circuit modeling, generalized signal graph, diacoptics methods.

DOI 10.14357/20718632220101

References

1. G. Kron 1972 Issledovanie slozhnyh shem po chastjam – diakoptika [Research of complex systems in parts - diacoptics], Moscow: Science Pubs 544 p.
2. H. Hepp 1974 Diakoptika i jelektricheskie cepi [Diacoptics and electrical circuits], Moscow: Mir Pubs 342 p.
3. B.V. Batalov, Ju.B. Egorov, S.G. Rusakov Osnovy matematicheskogo modelirovanija bol'shij integral'nyh shem na JeVM [Fundamentals of mathematical modeling of large integrated circuits on computers], Moscow: Radio i svjaz' Pubs 168 p.
4. V.I. Anisimov, V.N. Gridin 2009. Metody postroenija shem avtomatizirovannogo proektirovanija na osnove Internet-tehnologij i kompaktnoj obrabotki razrezhennyh matric [Methods of constructing computer-aided design diagrams based on Internet technologies and compact processing of sparse matrices]. Informacionnye tehnologii v proektirovanii i proizvodstve [Information Technology in Design and Manufacturing] 1: 3-7.
5. O.S. Kovalenko, V.M. Kurejchik 2012 Obzor problem i sostojanij oblachnyh vychislenij i servisov [Overview of cloud computing and services challenges and conditions] Izvestija JuFU. Tehnicheskie nauki [News of the SFU. Technical sciences] 7: 146-153.
6. V.N. Gridin, G.D. Dmitrevich, D.A. Anisimov 2011 Postroenie shem avtomatizirovannogo proektirovanija na osnove Web-tehnologij [Building Computer-Aided Design Diagrams Based on Web Technologies] Informacionnye tehnologii [information technologies] 5: 23-27.
7. D.A. Anisimov 2012 Metody postroenija shem avtomatizacii shemotehnicheskogo proektirovanija na osnove veb-servisov [Methods of constructing diagrams for automation of schematic design based on web services]. Izvestija SPbGJeTU «LJeTI» [News of St. Petersburg State Technical University "LETI"] 10: 56-61.
8. GOST 34.601-90 2009. INFORMACIONNAJA TEHNOLOGIJA. Kompleks standartov na avtomatizirovannye sistemy // Avtomatizirovannye sistemy stadii sozdanija. [INFORMATION TECHNOLOGY. Set of standards for automated systems // automated systems of the creation stage] Moscow: StandardinformPubs.
9. ISO/IEC 12207:2008. Systems and software engineering — Software life cycle processes
10. James Taylor Managing Information Technology Projects: Applying Project Management Strategies to Software, Hardware, and Integration Initiatives. Delhi: PHI Learning, 2004.
11. V.I. Anisimov 1977 Topologicheskij raschet jelektronnyh shem [Topological calculation of electronic diagrams] Leningrad: Energy Pubs 238 p.
12. V.N. Gridin, V.B. Mihajlov, L.B. Shusterman 2008 Chislenno-analiticheskoe modelirovanie radiojelektronnyh shem [Numerical and analytical modeling of electronic circuits], Moscow: Science Pubs 339 p.
13. B.A. Kalabekov, I.Ju. Lapidus, V.M. Malafeev 1990 Metody avtomatizirovannogo rascheta jelektronnyh shem v tehnikе svjazi [Methods of automated calculation of electronic circuits in communication technology], Moscow: Radio i svjaz' Pubs 272 p.

Gridin V. N. Design information technologies Center Russian Academy of Sciences, scientific director, Dr.Sci.Tech., professor, e-mail: info@ditc.ras.ru, Odintsovo, st. of the Marshala Biryuzova of 7a.

Anisimov V. I. Design information technologies Center Russian Academy of Sciences, chief researcher, Dr.Sci.Tech., professor, e-mail: info@ditc.ras.ru, Odintsovo, st. of the Marshala Biryuzova of 7a. Saint-Petersburg Electrotechnical University, Dr.Sci.Tech., professor, St. Petersburg, Professor Popov St. 5.

Vasilev S. A. Design information technologies Center Russian Academy of Sciences, candidate of technical sciences, senior Researcher, e-mail: info@ditc.ras.ru, Odintsovo, st. of the Marshala Biryuzova of 7a. Saint Petersburg Electrotechnical University, assistant, St. Petersburg, Professor Popov St. 5.