ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.031.6

Евгений Рафаилович Пантелеев

ФГБОУВО «Ивановский государственный энергетический университет имени В.И. Ленина», доктор технических наук, профессор кафедры программного обеспечения компьютерных систем, Россия, Иваново, телефон (4932) 26-98-60, e-mail: erp@poks.ispu.ru

Арман Арсенович Мукучян

ФГБОУВО «Ивановский государственный энергетический университет имени В.И. Ленина», магистрант кафедры программного обеспечения компьютерных систем, Россия, Иваново, e-mail: 15508@gapps.ispu.ru

Михаил Алексеевич Кузнецов

ФГБОУВО «Ивановский государственный энергетический университет имени В.И. Ленина», магистрант кафедры программного обеспечения компьютерных систем, Россия, Иваново, e-mail: 15502@gapps.ispu.ru

Алевтина Леонидовна Алыкова

ФГБОУВО «Ивановский государственный энергетический университет имени В.И. Ленина», кандидат технических наук, доцент кафедры программного обеспечения компьютерных систем, Россия, Иваново, телефон (4932) 26-98-60, e-mail: alla.ispu@yandex.ru

Метод формирования контекстной помощи пользователю компьютерного приложения в процессе решения прикладной задачи

Авторское резюме

Состояние вопроса. Применение программного обеспечения для решения прикладных задач вызывает проблему реализации необходимого пользователю действия при помощи интерфейса приложения. Частично она снимается в ходе изучения справочников и консультаций у разработчиков приложения. Однако справочники структурированы в контексте описания общего функционала приложения, а затруднения возникают в контексте состояния данных решаемой задачи. Консультации лишены этого недостатка, но дороги и не всегда оперативно доступны. Это стимулировало разработку компьютерных методов контекстной помощи, однако до сих пор задача формирования рекомендаций по выполнению запрошенной операции из текущего состояния данных приложения не решена. Цель исследования — сокращение временных затрат на получение помощи путем разработки модели представления знаний, обусловливающей действия пользователя контекстом данных приложения, и метода вывода рекомендаций на ее основе.

Материалы и методы. Модель сценария действий пользователя представлена в виде раскрашенной сети Петри. Это решение обосновано аналогией между сценариями действий пользователя и сценариями управления потоками работ (workflow), для представления которых аппарат сетей Петри давно

[©] Пантелеев Е.Р., Мукучян А.А., Кузнецов М.А., Алыкова А.Л., 2020 Вестник ИГЭУ, 2020, вып. 5, с. 64–76.

и успешно используется. Для топологического анализа сети Петри применена стратегия исчерпывающего поиска вглубину.

Результаты. Предложен метод формирования контекстных рекомендаций по выполнению запрошенной пользователем операции на основе модели сценария в виде раскрашенной сети Петри. Метод отличается применением топологического анализа сети Петри для построения множества альтернативных сценариев выполнения операции с последующей фильтрацией альтернатив в процессе пошагового исполнения рекомендованных действий. Метод апробирован при решении существенно интерактивных задач редактирования инженерных моделей.

Выводы. Предложенный метод обеспечивает получение контекстной помощи за одно обращение. При использовании для этой цели традиционных справочных файлов количество обращений определяется количеством возможных вариантов выполнения операции.

Ключевые слова: контекстная помощь, компьютерное приложение, сети Петри, топологический анализ

Evgeny Rafailovich Panteleev

Ivanovo State Power Engineering University, Doctor of Engineering Sciences (Post-doctoral degree), Professor of the Computer Systems Software Support Department, Russia, Ivanovo, telephone (4932) 26-98-60, e-mail: erp@poks.ispu.ru

Arman Arsenovich Mukuchan

Ivanovo State Power Engineering University, Master degree student of the Computer Systems Software Support Department, Russia, Ivanovo, e-mail: 15508@gapps.ispu.ru

Michael Alekseevich Kuznetzov

Ivanovo State Power Engineering University, Master degree student of the Computer Systems Software Support Department, Russia, Ivanovo, e-mail: 15502@gapps.ispu.ru

Alevtina Leonidovna Alykova

Ivanovo State Power Engineering University, Candidate of Engineering Sciences, Associate Professor of the Computer Systems Software Support Department, Russia, Ivanovo, telephone (4932) 26-98-60, e-mail: alla.ispu@yandex.ru

Method of context-dependent assistance for software user solving an applied task

Abstract

Background. When using software to solve the applied tasks, the problem to implement the action in need by means of user interface arises. Partly, this problem is solved by studying reference manuals and consultations with the application developers. However, reference manuals are structured in the context of overall application functionality while difficulties arise in the data state context of the task being solved. Consultations lack this flaw, but they are costly and not always available in time. This fact stimulated the development of computer-based methods of contextual help. Yet the task of development of recommendations for performing the requested operation from the current state of the application data has not been solved so far. The research is aimed to reduce the time to get help by developing a knowledge representation model that determines user actions across the application data context, and a method for deriving model based recommendations.

Materials and methods. The model of the user action scenario is presented in the form of a colored Petri net. This decision content is based on the analogy between user action scenarios and workflow scenarios, for which Petri nets notation has been successfully used for years. For the topological analysis of the Petri net, the strategy of exhaustive depth-first search was applied.

Results. The method of the contextual recommendations is proposed to execute the operation requested by the user based on a scenario model in the form of a colored Petri net. The method novelty is application of topological analysis of the Petri net to construct a set of alternative scenarios for performing the operation, followed by filtering alternatives in the process of stepwise execution of the recommended actions.

Conclusion. The suggested method provides context-dependent assistance in just one click. When using traditional reference manuals and files for the purpose, the number of clicks is determined by the number of options available to perform the operation.

Key words: context-dependent assistance, computer application, Petri nets, topological analysis

DOI: 10.17588/2072-2672.2020.5.064-076

Введение. Потенциал повышения эффективности решения прикладных задач, заложенный в специализированном программном обеспечении, часто не реализуется в полной мере из-за неумения пользователя транслировать известный ему алгоритм решения задачи в соответствующую этому алгоритму последовательность действий с элементами интерфейса приложения.

Использование традиционных ресурсов методического сопровождения не позволяет устранить это противоречие. Справочные руководства описывают общий функционал приложения, тогда как проблемы чаще всего возникают в связи с использованием этого общего функционала для решения конкретной задачи. Поэтому пользователь вынужден выстраивать цепочку действий, необходимых для решения задачи, трудоемким методом проб и ошибок. Например, в графическом редакторе схем электроснабжения системы EnergyCS [1] добавление ветви в схему предполагает, что на момент выполнения этой операции в схеме уже существует хотя бы один узел и этот узел выделен в качестве точки привязки ветви. Пользователю может потребоваться до трех обращений к традиционному справочному ресурсу, чтобы выяснить, как создать ветвь, как выделить узел и как создать узел. Такой подход часто оказывается неприемлемым, так как решение задач входит в круг производственных обязанностей и имеет жесткие временные регламенты. Кроме того, зачастую эмпирически построенная цепочка действий может оказаться неоптимальной, и многократное использование ошибочного стереотипа в будущем также повлечет за собой неоправданные затраты времени. Например, при добавлении ветви «Трансформатор» в редакторе схем электроснабжения системы EnergyCS нет необходимости создавать для этой ветви два узла привязки. Достаточно задать один на стороне высокого напряжения, второй узел будет создан системой автоматически. Хотя система не препятствует созданию второго узла вручную, такой вариант действий пользователя не является оптимальным. Консультации разработчиков программного обеспечения, конечно, решают эту проблему, но они не всегда мгновенно доступны и, как правило, оказываются на платной договорной основе. Поэтому становится очевидной актуальность разработки компьютерных моделей, увязывающих общий функционал приложения с контекстом его применения (определяется состоянием данных решаемой задачи), и механизмов построения сценариев рекомендованных действий на этой основе. Попытки разработки подобных моделей и методов их интерпретации как в качестве инструмента обучения решению прикладных задач вообще, так и в качестве инструмента, обучающего использованию для этой цели специализированного программного обеспечения, предпринимались многими исследователями.

В качестве примера обучения решению прикладных задач можно привести интеллектуальную обучающую систему SlideTutor для диагностики заболеваний по визуальным признакам [2] с использованием форматов представления предметных знаний (онтологий и правил продукции) и механизмов их интерпретации, делающих этот подход потенциально применимым и для других (не медицинских) областей применения. Контекстная помощь SlideTutor предоставляется в виде пошаговых инструкций и диагностики ошибочных шагов студента. К недостаткам этого подхода можно отнести необходимость совместного использования двух различных формализмов представления знаний для представления их декларативной (онтологии) и процедурной (продукции) составляющих, а также недостаточную глубину предоставляемых системой инструкций (один шаг), обусловленную спецификой интерпретации правил продукции.

В качестве примера системы, реализующей обучение контекстному использованию специализированного программного обеспечения, можно привести систему распознавания стратегии действий пользователя с элементами интерфейса приложения [3]. Под распознаванием стратегии понимается ее сопоставление с эталоном выполнения операции, представленным в виде дерева. Листьям дерева сопоставлены базовые действия, а внутренним вершинам - построенные на их базе операции. Дополнительно на этой древовидной структуре специфицируются ограничения на последовательность выполнения операций во времени. Процедуру распознавания реализует специальный алгоритм, который отображает линейную последовательность действий пользователя на один из путей в эталонном дереве. К недостаткам этого подхода следует отнести моделирование динамической системы (стратегии действий) в виде статической древовидной структуры. Это потребовало расширения базового формализма (дерева) связями, отражающими временную зависимость операций, и построения специализированного алгоритма распознавания на этой гибридной структуре.

Обсуждение этих примеров позволяет сделать вывод, что задача разработки компьютерных моделей, увязывающих общий функционал приложения с контекстом его применения, и механизмов построения сценариев рекомендованных действий на этой основе, сохраняет свою актуальность. Основанием для подобного утверждения является тот факт, что известные решения используют формализмы представления моделей, которые, несмотря на их популярность, не отражают динамическую природу задачи построения контекстной помощи, а построенные на их базе механизмы вывода либо ограничивают глубину подсказок, либо являются узкоспециализированными.

Цель исследования — повышение эффективности контекстной помощи, выражающееся в сокращении числа обращений к справочному ресурсу при решении прикладной задачи в среде специализированного программного обеспечения путем разработки модели представления знаний, обусловливающей действия пользователя контекстом данных приложения, и метода вывода рекомендаций на ее основе.

Материалы и методы. Обсуждаемый ниже подход к формированию контекстных рекомендаций пользователю специализированного программного обеспечения основан на следующих допущениях.

- 1. Решаемая при помощи приложения задача имеет конечное множество дискретных состояний, каждое из которых характеризует определенное качественное состояние данных этой задачи (ее контекст).
- 2. Контекст задачи определяет подмножество доступных в этом контексте действий, которые пользователь может активировать при помощи элементов интерфейса приложения.
- 3. Выполнение действия, активированного пользователем, меняет контекст приложения.

Из перечисленных допущений следует, что модель, лежащая в основе метода вывода контекстных рекомендаций, должна учитывать дискретную природу контекста, наличие условной связи контекста и доступных в этом контексте действий и изменение контекста в результате выполнения любого из доступных действий. Этим требованиям удовлетворяют модели, построенные в формате правил продукции, фреймов или сетей Петри [4]. Однако из приведенных альтернатив наиболее предпочтительной является последняя. Это объясняется тем, что формализм сетей Петри построен на строгом математическом основании и имеет наглядную графическую интерпретацию, которая описана далее с некоторыми упрощениями. Сеть Петри (СП) – это направленный двудольный граф. Вершины одного из подмножеств называются позициями (традиционно изображаются в виде окружности или эллипса), они могут быть использованы для хранения контекста, обусловливающего возможность выполнения действий. Значение контекста определяет разметка (маркировка) позиций: каждой позиции сопоставляются маркеры, совокупность которых определяет дискретное состояние данных приложения. Вершины второго подмножества – переходы (изображаются в виде прямоугольников) – могут быть интерпретированы как действия. Позиции и переходы СП связывают дуги. Дуга, направленная из позиции в переход, делает эту позицию входной по отношению к данному переходу. Дуга, направленная из перехода в позицию, делает позицию выходной по отношению к переходу. Динамический аспект СП определяют аксиомы активности и срабатывания переходов. Переход называется активным, если во всех его входных позициях присутствуют маркеры. Из нескольких активных в данном состоянии переходов один может сработать, в результате чего маркеры изымаются из входных позиций и помещаются в выходные позиции, моделируя тем самым изменение контекста задачи в результате действия пользователя.

Приведенное описание соответствует «плоской» черно-белой СП. Все переходы «плоской» сети находятся на одном уровне иерархии. Учитывая размерность модели, связывающей контекст задачи с возможными в этом контексте действиями, имеет смысл обсуждать СП иерархической струк-

туры (ИСП). Элементами, образующими иерархию, являются переходы. Внутренние вершины этой иерархии называются подстановочными переходами. Подстановочный переход моделирует операцию, для выполнения которой требуется несколько действий с элементами интерфейса приложения. Например, операция добавления ветви в уже упоминавшемся графическом редакторе схем EnergyCS требует выбора узла подключения, выбора типа ветви и отрисовки ее сегментов. Подстановочные переходы раскрываются в подсеть СП, топологически идентичную подстановочному переходу по множествам входных и выходных позиций. На уровне «листьев» иерархии ИСП превращается в «плоскую» СП, переходы которой соответствуют элементарным действиям пользователя с интерфейсом приложения. К таким действиям относится, например, выбор узла в редакторе схем EnergyCS. Маркер черно-белой сети можно интерпретировать как признак выполнения условия, относящегося к состоянию данных. Однако существуют раскрашенные (цветные) СП, маркеры которых можно рассматривать как типизированные переменные и структуры данных. Такая возможность, учитывая наличие параметризованных условий (например, тип ветви в редакторе схем EnergyCS), также должна быть представлена в модели. Поэтому далее будет обсуждаться модель, построенная на базе раскрашенной иерархической сети Петри (РИСП).

В пользу выбора СП в качестве формата представления модели сценариев говорит и тот факт, что СП давно и успешно используются для компьютеризации управления бизнес-процессами [5-9] путем создания систем управления потоками работ workflow management systems (WFMS). B рамках WFMS задания моделируются переходами СП, условия их выполнения разметкой позиций, что с точностью до обозначений совпадает с предлагаемым подходом. Отличие заключается только в целях использования модели. Если в системах управления потоками работ модель инициирует реальные процессы, то в системе контекстной помощи реальные действия пользователя инициируют смену состояния модели.

Кроме уже упомянутых преимуществ использования аппарата СП для представления моделей сценариев, необходи-

также использование MO отметить XML-формата для долговременного хранения моделей. Например, в [9] описан XML Color Petri Net (XCPNL). Это XML-схема, в которой описаны все разрешенные теги языка. Форма XML позволяет использовать для парсинга XML-документов существующие библиотеки программ. В качестве инструментальной основы разработки и отладки моделей сценариев в [9] использовалось приложение CPN Tools [10]. Для интерпретации действий пользователя на модели сценариев было разработано собственное программное обеспечение, которое основывается на описанной выше аксиоматике.

Принципиально важным моментом в создании системы является метод интерпретации запроса помощи для выполнения заданной пользователем операции из текущего контекста задачи. Так как в результате выполнения этой операции модель сценариев должна перейти в некое новое качественное состояние, для поиска возможных путей перехода РИСП из текущего состояния в новое можно воспользоваться стандартным аппаратом анализа достижимости разметок [10]. Однако процедура анализа достижимости имеет экспоненциальную вычислительную эффективность и потому применима только для СП небольшого размера. В [11] обсуждается альтернативный подход к решению этой задачи, предполагающий инвертирование (изменение направления дуг) сети. Но если для черно-белых сетей Петри задача инвертирования успешно решена, то для РИСП ее решение, помимо инверсии дуг, сопряжено с необходимостью инвертирования условий на дугах и значений цветных маркеров. Решение этих проблем сопряжено со значительными теоретическими сложностями, поэтому [11] носит исследовательский характер и ее результаты имеют преимущественно академический интерес. Их практическое применение ограничено описанными шаблонными структурами. Таким образом, разработка метода интерпретации запроса помощи для выполнения заданной пользователем операции из текущего контекста задачи сохраняет свою актуальность.

Результаты. Основным результатом исследования является метод интерпретации запроса контекстной помощи на модели сценариев в формате РИСП, из описа-

ния которых исключены ингибированные дуги. Данное ограничение упрощает разработку метода и при этом не является существенным, так как в любом случае для фрагмента РИСП, содержащего ингибированные дуги, можно построить эквивалентный фрагмент, который таких дуг не содержит, путем явного введения противоположного состояния — состояние Off (рис. 1). Можно заметить, что поведение этих двух фрагментов сети идентично: после срабатывания перехода В становится активным переход А.

Созданный в среде CPN Tools пример модели сценария добавления ветви в схему электроснабжения в графическом редакторе схем EnergyCS приведен на рис. 2. Показанный здесь фрагмент РИСП связан с остальной частью модели через входные In и выходные Out позиции, выделенные двойным контуром. Наряду с обычными переходами, соответствующими элементарным действиям пользователя, в модели представлены так называемые подстановочные переходы, выделенные двойным контуром. Подстановочные переходы в контексте решаемой задачи соответствуют операциям, требующим выполнения нескольких элементарных действий и/или операций. Например, подстановочный переход Draw Branch (нарисовать ветвь) раскрывается произвольным количеством действий по созданию отдельных сегментов ветви. В целом логика приведенного сценария сводится к следующему. Создание ветви требует выполнения определенных предусловий (в схеме имеются узлы nodes, один из которых выбран (Node Selected), и никакая другая операция в данный момент не выполняется – Empty State). Если предусловия выполнены, пользователь должен выбрать тип ветви (с одной или двумя точками подключения). Затем для узла типа «Шина» (Bar), имеющего протяженность, выбирается точка привязки ветви к этому узлу. Для узла типа «Точка» (Point) это действие не требуется. На заключительном этапе выполняется операция отрисовки ветви в виде произвольного количества смежных горизонтальных и вертикальных участков, по завершении которой формируются пост-условия выполнения операции «Создание ветви». Ветвь добавляется к множеству Branches ветвей схемы, устанавливаются флаги отсутствия выполняемой операции Empty State и отсутствия выделенного узла No Node Selected. Для реализации этой логики модель использует типизированные переменные (цветные маркеры), условия на дугах и переходах, что приближает возможности графической модели к возможностям алгоритмических языков.

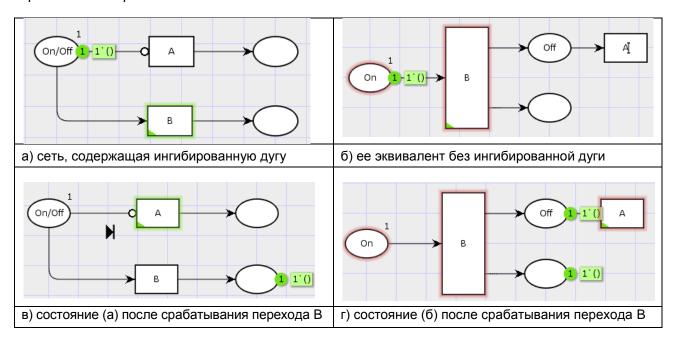


Рис. 1. Исключение ингибированных дуг

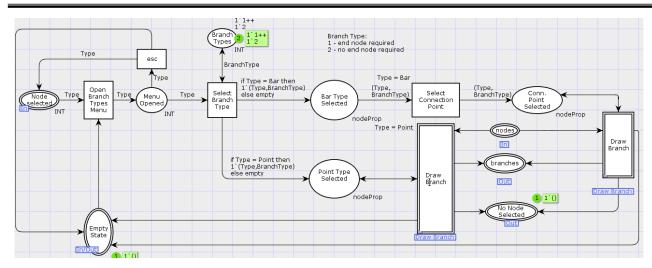


Рис. 2. Модель сценария создания ветви в формате РИСП

Запрос пользователя осуществляется в форме указания операции, которую он хотел бы выполнить из текущего качественного состояния данных (контекста) задачи. Например, пользователь хочет узнать, как выполнить операцию добавления ветви. Текущий контекст ее выполнения (типа «в схеме присутствуют узлы и ни один из них не выбран») устанавливается автоматически по состоянию разметки РИСП. Контекстная помощь предоставляется в форме инструкций по осуществлению действий с интерфейсом приложения, последовательность которых приводит к выполнению запрошенной операции.

Метод интерпретации запроса базируется на допущении, что состояние модели сценариев, в которое она переходит в результате выполнения запрошенной операции, достижимо из текущей разметки. Это допущение не ограничивает общность метода, так как обратное означало бы, что в приложении, для которого построена модель сценариев, существуют операции, активация которых при помощи интерфейса невозможна. Иначе говоря, приложение ограничивает доступ пользователя к своему функционалу, что, безусловно, не должно иметь места в действительности.

Анализ представленной на рис. 2 модели показывает, что переход из текущего контекста в контекст, соответствующий выполненной операции, допускает множество альтернативных вариантов действий, из которых пользователь должен выбрать единственный. Поэтому метод интерпретации запроса включает два последовательных этапа. Сначала строится множество допустимых вариантов, представленное подсетью РИСП, из которого затем действиями пользователя выделяется единственный вариант. Научный интерес представляет первый этап предлагаемого метода, основанный на инвертировании сети Петри. Отличительной особенностью подхода к реализации инверсии является игнорирование раскраски РИСП, обоснованное следующими допущениями:

- 1) топология РИСП отражает вариативность виртуальных сценариев;
- 2) раскраска РИСП ограничивает вариации сценариев.

Таким образом, построение множества допустимых вариантов может быть выполнено путем анализа топологии РИСП без учета раскраски и выражений на дугах, а раскраска – использована для детерминирования вариативного сценария действиями пользователя.

Процедура топологического анализа строит И/ИЛИ дерево [13] альтернативных вариантов выполнения запрошенной операции, которые затем используются для управления последовательностью срабатываний переходов (действий, необходимых для выполнения операции). Корнем И/ИЛИ дерева (И-вершиной) является инвертированный подстановочный переход, соответствующий запрошенной операции, внутренними узлами - промежуточные переходы (И-вершины) и позиции (ИЛИвершины), листьями - позиции, которые содержат маркеры. Для построения И/ИЛИ дерева использована стратегия исчерпывающего поиска вглубину, комбинаторная сложность которой уменьшается за счет иерархической организации РИСП.

Введем обозначения и определения, необходимые для описания алгоритма построения дерева:

- *P* множество позиций РИСП;
- Т множество переходов РИСП;
- текущая вершина И/ИЛИ дерева $n \in P \cup T$ имеет тип «И», если $n \in T$, иначе n имеет тип «ИЛИ», tn вершина И/ИЛИ дерева, смежная n;
- I множество вершин-предшественниц n;
- O множество вершин-последователей n;
- *V* множество переходов и позиций СП, которые были посещены в ходе рекурсивного построения ветви дерева. Узлы из *V* во избежание зацикливания далее не рассматриваются при построении ветви;
- *N* множество необходимых позиций, используется для проверки условия выполнения перехода. Под необходимыми понимаются позиции, в которых в процессе выполнения запрошенной операции обязательно должен оказаться маркер. Если в процессе построения дерева в позиции оказывается маркер, она удаляется из *N*;
- В множество заблокированных позиций. Заблокированными помечаются позиции с маркерами, которые связаны с рассматриваемым переходом входной дугой. Позиции из В, во избежание конфликта при передаче маркера, далее не рассматриваются в процессе построения дерева. Исключение составляют позиции, связанные с переходом двунаправленной дугой, так как после срабатывания перехода маркер будет возвращен в позицию.

Ниже описана рекурсивная функция построения дерева. СП позволяют описывать параллельные процессы, причем ветвление происходит из позиций, а синхронизация происходит в переходах, поэтому алгоритм построения И/ИЛИ дерева разбит на 2 блока — обработка позиций и обработка переходов. Если рассматриваемый на данном уровне рекурсии узел принадлежит V или B, функция возвращает null, иначе узел добавляется в множество V.

Обработка позиции начинается с проверки достижения листа (наличия в позиции маркера). В случае его отсутствия позиция удаляется из множества *N*, построение дерева запрашивается для всех вершин из *I*. Ненулевые результаты образуют множество потомков ИЛИ-вершины, а в

списке N остаются позиции, не получившие маркер.

При рассмотрении перехода все входные позиции добавляются в список N(условие срабатывания перехода СП), для всех позиций из / запрашивается построение дерева, предварительно В дополняется оставшимися узлами из / (за исключением узлов из $I \cap O$), полученные результаты запоминаются в буферный список. Дочерними tn становится множество результатов, для которых пересечение оставшихся N с Iпусто (множество таких действий, в результате которых все входные позиции имеют маркеры). Если такие множества найдены, из *N* вычитается *I*.

Ниже приведен псевдокод:

```
ЕСЛИ n \in V \lor n \in B ВЕРНУТЬ null КОНЕЦ ЕСЛИ V = V \cup \{n\} ЕСЛИ n \in P N = N \setminus \{n\} ЕСЛИ n содержит маркер ВЕРНУТЬ tn КОНЕЦ ЕСЛИ ДЛЯ КАЖДОГО i \in I tn. child = tn. child \cup Построение дерева \{i, N, V, B\} Оставить в N позиции без маркера
```

```
КОНЕЦ ЦИКЛА ВЕРНУТЬ tn КОНЕЦ ЕСЛИ ECЛИ ECЛИ
```

 $N = N \cup I$ ДЛЯ КАЖДОГО $i \in I$ Добавить в В остальные входные позиции $buffer = buffer \cup$

Построение дерева $\{i, N, V, B\}$ КОНЕЦ ЦИКЛА

Добавить в дочерние tn множество таких действий из buffer, в результате которых, все входные позиции имеют маркеры

ВЕРНУТЬ tn КОНЕЦ ЕСЛИ

Дерево альтернативных вариантов далее используется для выделения конкретной последовательности действий (И/ИЛИ-поддерева, обеспечивающего доставку маркеров из листьев во все выходные позиции корня, соответствующего запрошенной операции). И-вершины построенного методом поиска вглубину поддере-

ва образуют список необходимых действий. Позиция действия в списке определяется дисциплиной извлечения соответствующих переходов из стека (последним пришел – первым вышел), в который они попадают в процессе построения поддерева. Первыми в списке оказываются действия, доступные из текущего контекста. Порядок выполнения квазипараллельных действий детерминируется порядком обхода поддерева. Таким образом, множество возможных вариантов действий отображается в множество соответствующих списков. Текущие элементы этих списков соответствуют рекомендованным вариантам действий пользователя на текущем шаге. Интерпретация действия пользователя приводит к изменению текущего контекста приложения и состояния РИСП, отсечению неактуальных вариантов и продвижению по оставшимся спискам.

Описанный метод был апробирован на приложении – эмуляторе графического редактора схем EnergyCS. При открытии эмулятора EnergyCS запускается вспомогательное приложение, которое обрабатывает события графического редактора и

интерпретирует их как действия по активации переходов РИСП, обеспечивая, таким образом, изменение разметки РИСП в соответствии с контекстом приложения. Кроме того, приложение использует описанный выше метод для интерпретации запросов пользователя.

Рассмотрим запрос помощи «Построение ветви» в контексте «Существуют узлы (маркер в позиции nodes), ни один из них не выбран (маркер в позиции No Node Selected), нет активных действий (маркер в позиции Empty State)». Создается корень дерева – переход RequestedOp, входными позициями для которого являются branches и No Nodes Selected, и запрашивается построение дерева из нового узла. Фрагмент полученного дерева (не представлены элементарные действия перехода Draw Branch) показан на рис. 3. Для удобства сопоставления СП и построенного на нем дерева И/ИЛИ его узлы изображены в соответствии со стандартом визуализации СП: позиции представлены в виде овалов, переходы – в виде прямоугольников.

Отображение возможных вариантов действий на РИСП показано на рис. 4.

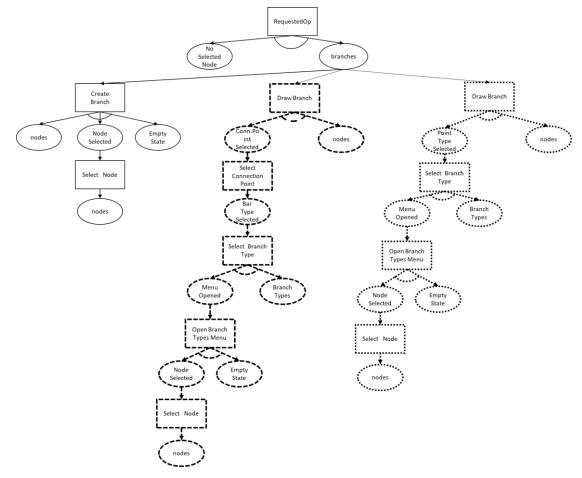


Рис. 3. Фрагмент дерева для запрошенной операции «Построение ветви»

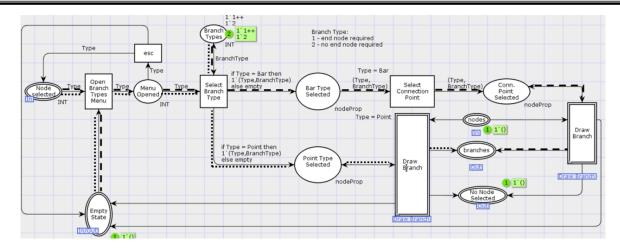


Рис. 4. Сопоставление СП и построенного на ней дерева

Пример работы комплекса представлен на рис. 5–8. Вспомогательное приложение показывает все возможные действия, которые способствуют достижению запрошенной цели. Для каждого действия возможен просмотр стандартной справки. По нажатию на соответствующую кнопку откроется необходимая страница. В пред-

ставленном примере есть 2 альтернативных варианта завершения построения ветви — выбор существующего узла и создание нового.

После выделения узла пользователю предлагается выбрать режим рисования ветви и тип ветви. Далее необходимо указать точку подключения на шине (рис. 6).

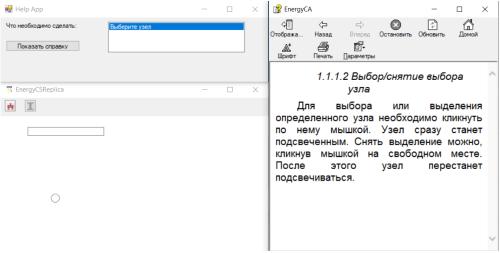


Рис. 5. Выделение узла

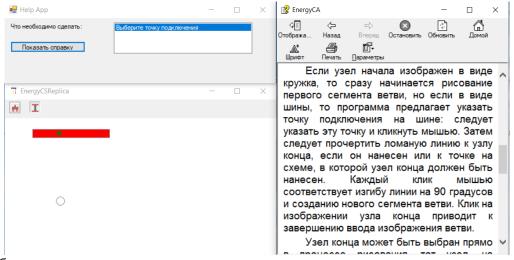


Рис. 6. Выбор точки подключения на шине

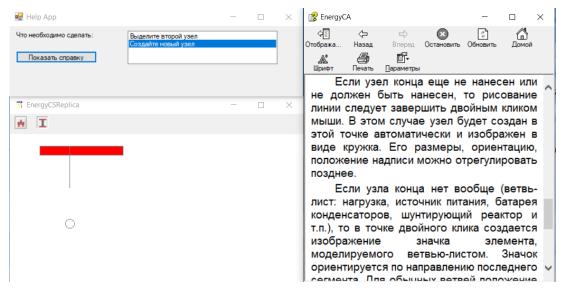


Рис. 7. Рисование ветви

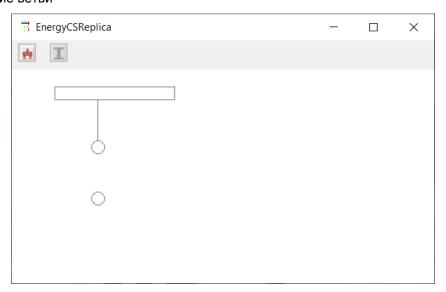


Рис. 8. Завершение построения ветви созданием нового узла

Выводы. Экспериментальное исследование программной реализации предложенного метода показало, что получение помощи о способах выполнения указанной пользователем операции из текущего контекста приложения доступно в результате единственного запроса. Таким образом, цель научного исследования — сокращение временных затрат на получение помощи — достигнута.

Достоверность результатов исследования подтверждается как обоснованным выбором и корректным применением математического аппарата сетей Петри, так и адекватностью полученных при помощи предложенного метода контекстных рекомендаций.

Полученные результаты могут быть использованы в процессе разработки и ме-

тодического сопровождения существенно интерактивного программного обеспечения, в частности приложений CAD/CAE. Приложения, которые могут использовать предложенный метод (приложенияклиенты), должны иметь открытый исходный код и быть построенными на использовании событийной связи между интерфейсом пользователя и обработчиками событий. Предложенный метод программно реализован в виде динамической библиотеки (сервера), которая предоставляет функционал по интерпретации сетей Петри и построению маршрутов достижения заданной маркировки. Интерфейс этой библиотеки предоставляет возможности передачи от клиента данных о совершенных пользователем действиях, а также исполнения запроса на формирование контекстной помощи. Для редактирования и отладки модели сценариев действий пользователя в формате РИСП доступно бесплатное приложение CPN Tools. Для создания .CHMфайлов справки можно использовать бесплатные инструменты (Microsoft HTML Help Workshop, Htm2Chm, плагины для Total Commander и т.п.). Для просмотра .CHMфайлов доступно стандартное средство просмотра, встроенное во все версии Microsoft Windows или сторонние программы просмотра.

Список литературы

- 1. Свид. 201ббб0884 Российская Федерация. Свидетельство о государственной регистрации в Реестре программы для ЭВМ. Система информационного моделирования электроэнергетических сетей для расчета установившихся режимов, токов короткого замыкания, потерь электрической энергии, технико-экономических показателей с отображением результатов на однолинейных схемах и схемах на картах местности (СИМЭС) / Н.Б. Ильичев, А.И. Кулешов, В.А. Серов и др. – M.: POCΠΑΤΕΗΤ, 2016.
- 2. **Crowley R., Medvedeva O., Jukic D.** SlideTutor: A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis // Proceedings of the 11th international conference on artificial intelligence in education. 2003. P. 157–164.
- 3. **Towards** collaborative intelligent tutors: Automated recognition of users' strategies / Y. Gal, E. Yamangil, S.M. Shieber, et al. // International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, 2008. P. 162–172.
- 4. **Котов В.Е.** Сети Петри. М.: Наука, 1984. 160 с.
- 5. **Van der Aalst W.M.P.** The application of Petri nets to workflow management // Journal of circuits, systems, and computers. 1998. Vol. 8, No. 01. P. 21–66.
- 6. **Zhao X., Choi Y.** Hierarchical Structuring of a Workflow Model in Petri-Net // ICEB. 2004. P. 117–121.
- 7. **INCOME/WF** A Petri net based approach to workflow management / A. Oberweis, R. Schätzle, W. Stucky, et al. // Wirtschaftsinformatik'97. Physica. Heidelberg, 1997. P. 557–580.
- 8. Verbeek H.M.W., Hirnschall A., van der Aalst W.M.P. XRL/Flower: Supporting inter-organizational workflows using XML/Petri-net technology // International Workshop on Web Services, E-Business, and the Semantic Web. Springer, Berlin, Heidelberg, 2002. P. 93–108.

- 9. **Александров Д.В., Фадин Д.Н.** Методика трансляции графического представления бизнес-процесса предприятия в раскрашенную сеть Петри // Вестник Тамбовского государственного технического университета. 2007. Т. 13, № 2.
- 10. **CPN** tools for editing, simulating, and analysing coloured Petri nets / A.V. Ratzer, L. Wells, H.M. Lassen, et al. // International Conference on Application and Theory of Petri Nets. Springer, Berlin, Heidelberg, 2003. P. 450–462.
- 11. **Bouali M., Barger P., Schon W.** Colored Petri Net inversion for Backward reachability analysis // IFAC Proceedings Volumes. 2009. Vol. 42, No. 5. P. 227–232.
- 12. **Tavana M.** Knowledge-based expert system development and validation with petri nets // Journal of Information & Knowledge Management. 2008. Vol. 7, No. 01. P. 37–46.
- 13. **Нильсен Н.** Искусственный интеллект: методы поиска решений: пер. с англ. М.,1973.

References

- 1. Il'ichev, H.B., Kuleshov, A.I., Serov, V.A., Shershnev, K.G., Il'ichev, A.N., Eliceeva, E.N. Sistema informatsionnogo modelirovaniya elektroenergeticheskikh setey dlya rascheta ustanovivshikhsya rezhimov, tokov korotkogo zamykaniya, poter' elektricheskoy energii, tekhniko-ekonomicheskikh pokazateley s otobrazheniem rezul'tatov na odnolinevnykh skhemakh i skhemakh na kartakh mestnosti (SIMES) [Information modeling systems for electric power networks for calculations of steady-state modes, short-circuit current, energy losses, technical and economic indicators with the results displayed on single-line diagrams and diagrams on terrain maps]. Svidetel'stvo o gosudarstvennoy registratsii v Reestre programmy dlya EVM no. 201ббб0884 [Certificate of state registration in the Register of computer programs no. 2016660884]. Moscow: ROSPATENT, 2016.
- 2. Crowley, R., Medvedeva, O., Jukic, D. SlideTutor: A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis. Proceedings of the 11th inter-national conference on artificial intelligence in education, 2003, pp. 157–164.
- 3. Gal, Y., Yamangil, E., Shieber, S.M., Rubin, A., Grosz, B.J. Towards collaborative intelligent tutors: Automated recognition of users' strategies. International Conference on Intelligent Tutoring Systems, Springer, Berlin, Heidelberg, 2008, pp. 162–172.
- 4. Kotov, V.E. *Seti Petri* [Petri Nets]. Moscow: Nauka, 1984. 160 p.
- 5. Van der Aalst, W.M.P. The application of Petri nets to workflow management. Journal of circuits, systems, and computers, 1998, vol. 8, no. 01, pp. 21–66.

- 6. Zhao, X., Choi, Y. Hierarchical Structuring of a Work-flow Model in Petri-Net. ICEB, 2004, pp. 117–121.
- 7. Oberweis, A., Schätzle, R., Stucky, W., Weitz, W., Zimmermann, G. INCOME/WF–A Petri net based approach to workflow management. Wirtschaftsinformatik'97, Physica. Heidelberg, 1997, pp. 557–580.
- 8. Verbeek, H.M.W., Hirnschall, A., van der Aalst, W.M.P. XRL/Flower: Supporting interorganizational work-flows using XML/Petri-net technology. International Workshop on Web Services, E-Business, and the Semantic Web, Springer, Berlin, Heidelberg, 2002, pp. 93–108.
- 9. Aleksandrov, D.V., Fadin, D.N. Metodika translyatsii graficheskogo predstavleniya biznesprotsessa predpriyatiya v raskrashennuyu set' Petri [Method of translating a graphical representation of an enterprise's business process into a colored Petri net]. Vestnik Tambovskogo gosudarstven-

- nogo tekhnicheskogo universiteta, 2007, vol. 13, no. 2.
- 10.Ratzer, A.V., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K. CPN tools for editing, simulating, and analysing coloured Petri nets. International Conference on Application and Theory of Petri Nets, Springer, Berlin, Heidelberg, 2003, pp. 450–462.
- 11.Bouali, M., Barger, P., Schon, W. Colored Petri Net inversion for Backward reachability analysis. IFAC Proceedings Volumes, 2009, vol. 42, no. 5, pp. 227–232.
- 12.Tavana, M. Knowledge-based expert system development and validation with petri nets. Journal of Information & Knowledge Management, 2008, vol. 7, no. 01, pp. 37–46.
- 13.Nil'sen, N. *Iskusstvennyy intellekt: metody poiska resheniy* [Artificial intelligence: methods of finding solutions]. Moscow, 1973.

ВЕСТНИК ИВАНОВСКОГО ГОСУДАРСТВЕННОГО ЭНЕРГЕТИЧЕСКОГО УНИВЕРСИТЕТА

Выпуск 5

Издание зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций. Свидетельство о регистрации ПИ N 77-13527 от 20.09.02 г.

Подписано в печать 14.10.2020. Выход в свет 31.10.2020. Формат $60x84 \frac{1}{8}$. Усл. печ. л. 8.83. Уч.-изд. л. 9.62. Тираж 100 экз. Цена свободная. Заказ

Ивановский государственный энергетический университет, 153003, г. Иваново, ул. Рабфаковская, 34. Типография ООО «ПресСто», 153025, г. Иваново, ул. Дзержинского, 39, оф. 307.