УДК 621.311

РАЗВИТИЕ ТЕОРЕТИЧЕСКИХ ОСНОВ И СОЗДАНИЕ МЕТОДА АВТОМАТИЧЕСКОГО ФОРМИРОВАНИЯ АДАПТИРУЕМОЙ МОДЕЛИ ЭЛЕКТРОЭНЕРГЕТИЧЕСКОГО ОБЪЕКТА

Заболотный И.П.
Донецкий национальный технический университет
ivp@elf.dgtu.donetsk.ua

The concept and method of build-up of a dynamic mock-up on a basis to introducings about plant of control skilled staff designed. Usage of a method allows to adapt model of plant of control as for different problems of control of local plant of an electrical power system, and for an actual situation.

Основная идея работы заключается в разработке функционального проблемно-ориентированного топологического метода, обеспечивающего автоматическое создание многофункциональной адаптированной модели объекта электроэнергетической системы на основе графического изображения схемы электрических соединений.

Важнейшим этапом в процессе решения технологической задачи по управлению электроэнергетическим объектом является создание его математической модели. В процессе изменения условий работы электрической сети из-за изменения топологии, изменения процесса или режима работы требуется создание модели, отражающей эти изменения. Таким образом, обоснование решения по управлению рабой электрической системы основано на использовании динамической модели. Степень изменения модели зависит от вида и длительности возмущения, характера протекания режима, эффективности работы систем управления и отражается либо в параметрах расчетной схемы замещения, либо в структуре расчетной схемы замещения и параметрах, а может потребовать изменения математических и моделей отдельных элементов и режима. Например, необходимость выполнения только переключений может привести не только к необходимости уточнения топологии сети и параметров схемы замещения, но и изменению математического описания. Примером такой ситуации может быть отключение из-за нарушений режима отдельных машин переменного тока с последующим включением. Отключение напряжений статора Ud, Uq приводит не к отключению генератора, а к короткому замыканию на зажимах. Отключение же от узла токов генератора вызывает в генераторе процессы, которые реально не существуют. Правда, в остальной сети при отключении токов генератора подобие сохраняется, однако при следующих операциях, таких как повторное включение генератора, процессы уже правильно отражаться не будут. Решение задачи отыскания места повреждения на воздушных линиях по данным параметров режима короткого замыкания чувствительно к изменению распределения токов нулевой последовательности в сети. Поэтому разземление или заземление нейтрали трансформаторов подстанций на ответвлениях от линии, либо заземление одной из цепей двухцепной линии и др. может привести к существенным ошибкам при решении задачи.

Таким образом, сложность протекания процессов в энергетических объектах приводит к тому, что наиболее полные математические модели представляют собой системы дифференциальных уравнений в частных производных высокого порядка, которые необходимо модифицировать. Такое описание является инструментом исследователя и неадекватно задачам автоматизированной системы управления. Модель должна соответствовать тем представлениям об объекте управления, которыми обладает квалифицированный персонал. Принципиально важным является то, что степень подробности и точности определяется не характеристиками объекта управления, а закономерностями его отражения в сознании персонала. Поэтому в [1-3] и в настоящей работе реализуется концепция построения автоматизированной системы управления из других посылок.

На основании анализа деятельности персонала сформулируем ряд положений, характеризующих объект управления и конструктивных для построения его динамической модели:

1. Образ объекта имеет иерархическую структуру, состоящую из нескольких уровней. Верхний уровень – модель объекта управления в целом. Каждый объект имеем графический образ. Нижний уровень – модель подсистемы объекта управления.

2. Образ объекта характеризуется динамически меняющейся структурой, в которую входят необходимые для решения технологической задачи модели подсистемы.

3. Следует различать влияния, отражающие физические законы протекания процессов и возникающие как следствие функционирования устройств автоматического регулирования, защиты и автоматики.

4. Отклонения параметров могут произойти лишь вследствие какого-то события. Все изменения происходят в дискретные моменты времени, соответствующие некоторым событиям.

5. Ряд элементов объекта управления может находиться в состоянии «включен» либо «отключен». Все элементы могут находиться в состоянии «находится в ремонте», «исправен», «поврежден», «отремонтирован» и

т.д.

6. Ряд элементов имеют строгие взаимосвязи, определяющие реакции на определенные события. Обычно это механические, технологические и функциональные блокировки.

7. Использование теоретико-множественных понятий. Для локального объекта это множество регистрируемых параметров и его подмножества. К ним относятся параметры релейного типа; параметры, контролируемые регуляторами, устройствами защиты и автоматики; параметры, по пороговым значениям которых работают автоматические блокировки; исполнительные органы регуляторов, блокировок, релейной защиты. Параметры увязаны с режимами работы и состоянием элементов. Это граничные параметры; параметры, связанные соотношениями баланса; параметры, соответствующие оптимальному режиму; включенные элементы; отключенные элементы; возмущения. Подмножества отражают характеристики элементов локального объекта; соответствуют свойствам параметрам в реализованной модели; соответствуют текущему состоянию элементов релейного типа; соответствуют вектору возмущений.

8. При анализе установившихся режимов учет устройств регулирования и состояние блокировок выполняется итерационным путем с уточнением вектора возмущения, уставок регуляторов с учетом ограничений на изменение, работы блокировок на каждой итерации.

9. Создание системы поддержки принятия решения оперативным персоналом.

В [4] выделены две принципиально различные формы операторской деятельности: выполнение известного алгоритма действий; деятельность целеустремленного оператора, осуществляющего процедуру выбора. Вторая форма связана с принятием решений, являющихся критическим моментом оперативной деятельности. Реализация второй формы требует интеграции фаз разделенной интеллектуальной оперативной деятельности: контроля и идентификации состояния объекта управления, диагностика неисправности, разработка цели, планирование действий, оценка и выбор альтернативы. Эти образующие деятельности органично входят друг в друга, но у каждой свои характерные отличия, общими же являются знания по объекту управлению.

Поскольку объект управления представлен совокупностью параметров с заданным отношением принадлежности и взаимосвязи, то образующие деятельности могут быть представлены лишь как различные формы преобразования, анализа и синтеза информационного потока. Причем упор делается на общность, универсальность приемов, используемых персоналом энергетических объектов. Стремление к общности находит отражение и в форме представления декларативной информации, и в технической реализации системы управления. Тот же подход применяется и к моделям образующих деятельности, а любые логические построения должны быть в случае необходимости полностью объяснены в терминах, не выходящих за пределы профессиональных знаний оператора.

Следует подчеркнуть, что эффективность реализация первой формы операторской деятельности в конкретных ситуациях также требует уточнения отдельных этапов типовых моделей деятельности.

Взаимосвязанный набор моделей образующих деятельности, позволяющий осуществлять решение любой оперативной задачи, реализуемой системой поддержки решений.

10. Одной из образующих деятельности персонала является диагностика неисправности, которая должна, как правило, дать ответ на вопросы: в чем проявляется неисправность; какой элемент вышел из строя; каков прогноз развития процесса; в чем проявляется влияние неисправности на свойства объекта управления и качество технологического процесса.

Способность персонала определить изменения в объекте управления по ограниченному числу признаков (параметров), которыми характеризуется объект управления, основана на адаптируемых под ситуацию моделях и на конкретных приемах обработки информации.

Изменения в объекте управления проявляются через отклонения параметров. Исходя из принятого событийного подхода (п. 5), причиной любых отклонений являются возмущения, которые могут быть отнесены либо к управляющим воздействиям, которые наносятся либо непосредственно оператором, либо устройствами автоматики, либо к возмущениям, являющиеся следствием неисправности либо изменений внешней среды.

Таким образом, в зависимости от решаемой задачи управления возникает необходимость создания моделей для информационной поддержки решений персонала (информационная модель), модели для реализации коммутаций и оценки режимов, моделей диагностики, моделей работы релейной защиты и автоматики, упрощенных моделей электрических соединений для имитационного моделирования.

Создание модели релейной защиты включает в себя описание зон релейных защит элемента локального электроэнергетического объекта, условий срабатывания защит, выключателей, на которые воздействует защита. При этом формируются правила следующего типа «Если существует защита X типа Y и эта защита сработала, а также существует некий выключатель Z, причем X воздействует на его отключение, то выключатель Z отключается» [5]: $\forall x\ \forall y\ \forall z\ R(x,y) \wedge Q(x) \wedge S(z) \wedge T(x,z) \supset T(z)$,

где - R, Q, S, T - предикатные символы; x, y, z –переменные.

Так как очередность решения и количество задач разного направления в конкретной ситуации заранее определить невозможно, то возникает необходимость адаптации динамической модели.

В [1-2] приведены разработки по автоматизации создания на основе графического изображения модели режима электрической сети. Опыт использования показывает, что реализация не дает оптимального решения задачи при программировании. В [3] предлагается объектно-ориентированный подход к разработке программ анализа коммутационных схем электрических сетей. Объектно-ориентированное программирование основано на использовании достаточно сложного понятия объекта вместо простых понятий процедура/функция.

Однако, как показывает анализ, построение динамических моделей связано с обработкой внутренней базы чертежа, внешних баз данных, связанных с объектами чертежа, базами знаний, используемых в математических и логических моделях при решении разнообразных технологических задач, а при работе с базами данных четко видна пробуксовка идей ООП. Известно, что на смену технологии работы с БД файл-сервер пришли технологии клиент-сервер. Но при этом при правильном проектировании баз данных и приложений, работающих с ними, приходится иметь дело не с двумя уровнями, а с несколькими. К ним можно отнести таблицы со сложными связями между ними; хранимые процедуры реализации логики доступа к данным и их обработки; программные объекты, обеспечивающие механизмы интеллектуального отбора и редактирования данных пользователем, алгоритмом и другим программным объектом; алгоритм, обеспечивающий работу программы, использующий программные объекты для работы с БД; часть программы, обеспечивающей интерфейс взаимодействия с внешней средой. При этом объекты каждого уровня произвольным образом использовать любые другие объекты, находящиеся на данном уровне. Отсюда и возникают проблемы, среди которых можно отметить:

- отличия в методиках проектирования и реализации объектов разных уровней;
- необходимости введения не только значительного числа классов для описания объектов, но и сложных методов обработки для того, чтобы автоматически учесть изменения на одном уровне на других. При этом объекты образуют множество наборов, отдельные объекты могут входить в несколько наборов одновременно. Следует подчеркнуть, что сложности увязывания объектов растут значительно быстрее по сравнению с ростом количества объектов.

Анализ показывает, что дальнейшее развитие изложенного в [2] метода требует другого подхода к описанию объектов по сравнению с ООП. Несмотря на разницу в строении объектов их нужно не только разделить, что уже сделано в [2-3], на группы: по общности функционального назначения (что делают объекты) и (или) по сходству функциональных наборов (как они что-то делают), но и использовать такое деление для образования объектов другого вида. Под функциональным набором понимается перечень действий, выполняемых объектом как самостоятельно, так и по инициативе других программных объектов и событий. Все действия могут быть разбиты на последовательности.

Таким образом, мы имеем несколько совокупностей объектов, причем в каждой совокупности объекты функционируют сходным образом. При этом внутреннее и внешнее устройство объектов не принимается во внимание. Кроме стандартного функционального набора, каждый экземпляр из группы может иметь дополнительные функции, т.е. не накладываются никакие ограничения на объем и выполняемые действия.

Для обеспечения взаимодействия объекта с другими объектами используется набор параметров при полном отсутствии таких сущностей как процедуры, функции, методы. На рис. 1 приведена структурная схема метода формирования динамической модели. С помощью управляющих параметров устанавливается связь с пунктами меню, определяющими стратегию решения оператором, с промежуточными (текущими) результатами решения задачи, с обоснованиями модулей принятия решений на основе баз знаний.

Взаимодействие объектов основывается на использовании таблиц обмена данными. Требования

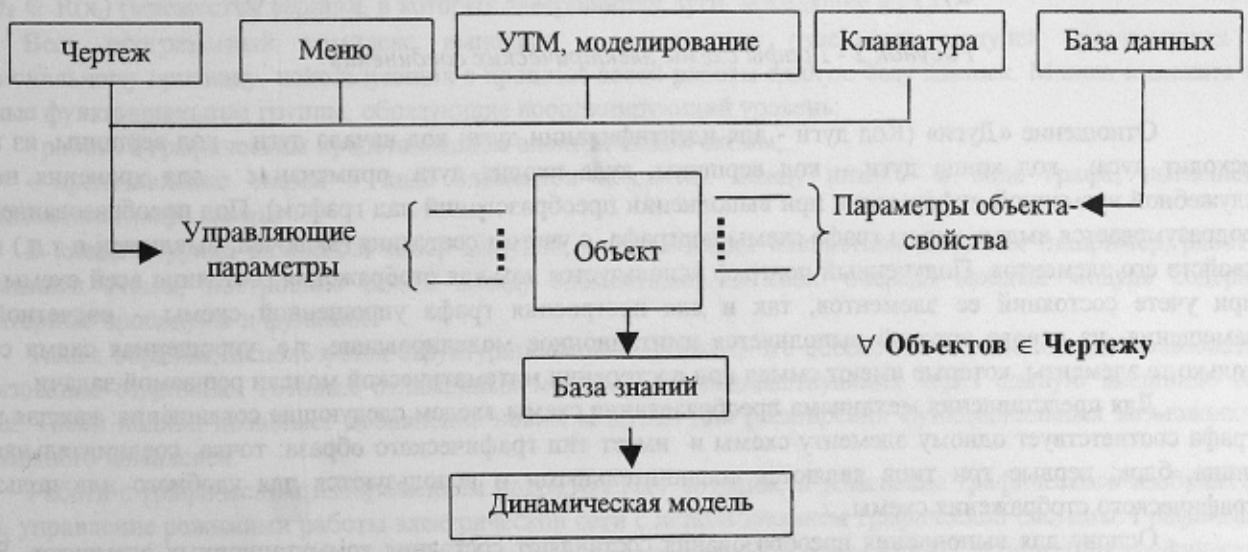

Рисунок 1 – Структурная схема метода формирования динамической модели

образования пространства имен и соглашений, полей, наборов хранимых процедур и т.д. частично описаны в [2]. Согласно [2] объект графического изображения имеет множество портов для соединения силовых элементов электрической схемы, подключения средств регулирования, измерения, подключения информационных окон. Параметры объектов позволяют определить тип; состояние; математическую модель; таблицы, описывающие свойства объекта; таблицы баз знаний; логические модели и т.д. Объекты реализуются в виде узлов и ветвей и согласно схеме соединения элементов электрической системы образуют граф соединений. На основании текущего состояния объекта и этапа решаемой задачи формируются подграфы графа соединений.

Таким образом, методы автоматического создания расчетных моделей реализованы автором в виде системы операций на графических изображениях электрических соединений элементов электрической системы. В результате обработки сложного графического изображения имеем граф сети. В качестве операционного базиса принимаются следующие бинарные операции над графами: объединение, пересечение, разность графов A и B; приращение графа A в графе B; замыкание графа A в графе B.

Если выбор пяти перечисленных выше операций над графами определяется спецификой рассматриваемого приложения, то содержание этих операций от данной специфики никак не зависит: операции определяются для любых графов. Назовем графом схемы множество $X = \{x_i\}$элементов этой схемы и введем множество пар $A = \{(x_i, x_j)\}$, указывающих, что между элементами x_i и x_j существует соединение (рис. 2, а).

Множество X называется множеством вершин или вершинами графа, а множество A - множеством дуг или дугами графа. Таким образом, граф определяется двумя множествами [6, 7] G = (X, A).

Полученный граф G является неориентированным графом, т.к. важно только само наличие соединения между элементами, а не направление соединения. Но для удобства работы граф лучше представить в виде ориентированного (рис. 2, б). Для этого каждая его неупорядоченная дуга заменяется парой упорядоченных дуг. Например, $(x_i, x_j) \Rightarrow (x_i, x_j)$ и (x_j, x_i).

Путь в ориентированном графе это последовательность дуг в котором конец каждой дуги есть начало следующей кроме последней. Пример: путь из x_1 в x_5 (x_1,x_2), (x_2,x_4), (x_4,x_5). Ориентированный граф называется сильно связным, если для любой пары вершин x_i и x_j существует путь их соединяющий. Правильно созданная схема не должна содержать «висящих» элементов, т.е. все элементы должны быть соединены между собой. Объединяя это с тем положением, что каждая неориентированная дуга заменяется двумя разнонаправленными ориентированными дугами, получаем что, полученный ориентированный граф является сильно связным графом [7]. Для хранения графа в памяти используются два отношения, схемы которых описаны ниже.

Отношение «Вершина» (Код поля –для идентификации вершины; наименование - в виде символьной кодовой строки, определяющей тип и параметры элементы схемы; примечание - для хранения некоторой служебной временной информации при выполнении преобразований над графом).

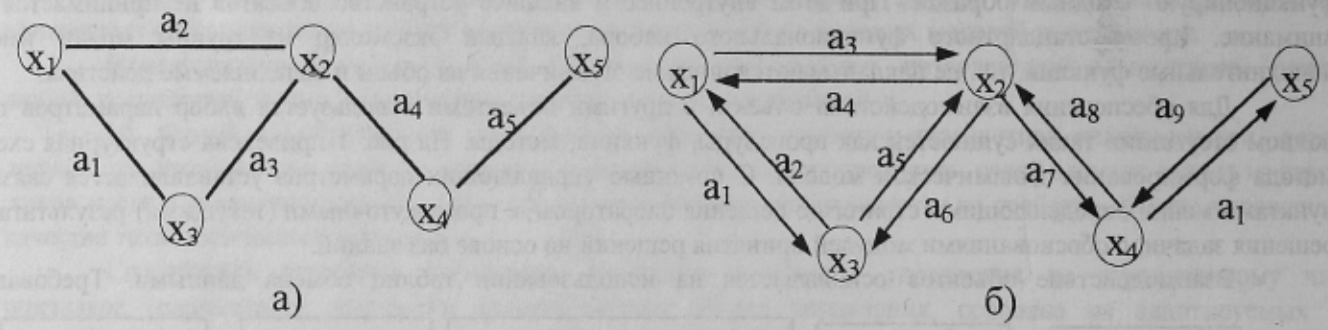

Рисунок 2 - Графы схемы электрических соединений

Отношение «Дуги» (Код дуги - для идентификации дуги; код начала дуги – код вершины, из которого исходит дуга; код конца дуги – код вершины, куда входит дуга; примечание - для хранения некоторой служебной временной информации при выполнении преобразований над графом). Под преобразованием схемы подразумевается выделение из графа схемы подграфа, с учетом состояния (включен, выключен и т.д.) и других свойств его элементов. Полученный подграф используется как для отображения состояния всей схемы в целом при учете состояний ее элементов, так и для построения графа упрощенной схемы – расчетной схемы замещения, на основе которой выполняется имитационное моделирование, т.е. упрощенная схема содержит только те элементы, которые имеют смысл при построении математической модели решаемой задачи.

Для представления механизма преобразования схемы введем следующие соглашения: каждая вершина графа соответствует одному элементу схемы и имеет тип графического образа: точка, соединительная линия, шина, блок; первые три типа являются соединительными и используются для удобного для пользователя графического отображения схемы.

Основу для выполнения преобразования составляют состояния коммутационных элементов. Выберем подмножество $K = \{k_i\}$ вершин, соответствующих коммутационным элементам схемы. Для любого k_i если

состояние «выключен», то исключить все исходящие из него дуги.

$$A' = A - H_i,$$

где H_i - множество дуг, исходящих из вершины k_i

Полученный граф G' = (X, A') уже может не обладать свойством сильной связности. Сильно связная (сильная) компонента графа G' это сильно связный подграф, который не содержится в любом другом сильном подграфе. Задача состоит в выделении сильных компонент, содержащих все элементы типа «источник энергии». При составлении графа состояние коммутирующего аппарата не учитывается.

Для выделения сильных компонент графа используется алгоритм, который приведен ниже.

1. Искомый подграф S = ().
2. Выбираем вершину x_i с типом «источник энергии» из графа G'.
3. Если такой вершины нет, то конец.
4. Находим множества $R(x_i)$ - вершины, которые можно достичь из x_i и $Q(x_i)$ - вершины, из которых можно достичь x_i
5. $S' = R(x_i) \cap Q(x_i)$ - будет сильной компонентой. Удаляем эти вершины из графа $G' = G' - S'$.
6. $S = S \cup S'$. Добавляем эти вершины к искомому подграфу S.
7. Если граф G' не пуст, то переход к 2.

Принимая во внимание особенности данного графа, если есть соединение в одну сторону, то всегда есть соединение и в другую сторону, кроме как для коммутационного оборудования. Поэтому, в вышеуказанном алгоритме, при пересечении множеств R и Q будут исключены коммутационные элементы, которые соединены с источниками энергии и имеют состояние «выключен». В этом случае, множество Q можно не определять. Теперь алгоритм сводится к поиску всех вершин, которые можно достичь из «источников энергии».

1. Искомый подграф S = ().
2. Выбрать множество вершин $C(x_i)$ графа G, что имеют тип «источник энергии».
3. Если $C(x_i)$ пусто, то выход
4. Для каждой вершины $x_j \in C(x_i)$
5. Выбрать множество вершин $R(x_j)$, в которых заканчиваются исходящие дуги вершины x_j, и которые не принадлежат искомому графу S.
6. Добавить вершины множества C к графу S
7. C = R
8. Переход к 3.

Дуги подграфа S могут быть получены удалением из множества A' дуг графа G' всех входящих и исходящих дуг вершин, не принадлежащих множеству S.

$$D = A' - (R(x_i) \cup Q(x_i)), \quad x_i \notin S$$

где $R(x_i)$ - множество исходящих дуг для вершины x_i, $Q(x_i)$ - множество входящих дуг для вершины x_i,

Чтобы выполнить проверку на исправность оборудования, необходимо определить, не существует ли в множестве S элемента, имеющего состояние «неисправен».

Чтобы получить упрощенный вариант схемы, из графа удаляются лишние вершины и дуги.

Для каждой вершины $x_i \in S$, если она не несет смысловой нагрузки (соединительная линия, шина, точка, коммутационный элемент), каждая входящая дуга (x_j, x_i) заменяется дугами из начальной точки x_j во все точки $r_k \in R(x_i)$ (множества вершин, в которых завершаются дуги, исходящие из x_i).

Весь программный комплекс выполнен в виде ряда отдельных модулей, разделенных по функциональному признаку, использующих в процессе своей работы единую базу данных. Можно выделить три основные функциональные группы, образующие координирующий уровень:

- работа с графическим представлением электрической схемы;
- представление схемы в виде элементов и связей между ними - в виде графа; выполнение преобразования графа схемы.

В каждой группе находится набор модулей, выполняющих конкретные действия (например, работа с фрагментами схемы, построение связей между элементами). В свою очередь каждый модуль содержит элементарные процедуры и функции.

Таким образом, используется структурированный подход, что обеспечивает удобное для пользователя использование отдельных готовых отлаженных модулей, взаимодействующих через единую внешнюю базу данных. Такой подход позволяет добавление новых модулей для расширения функциональных возможностей программного комплекса.

Работа с графическим изображением подразумевает создание и изменение графического изображения схемы, управление режимами работы электрической сети с использованием графической системы. Графическое изображение обычно несет очень большое количество информации. Как указано в [2] детализация графического изображения зависит от вида решаемой технологической задачи. Управление детализацией выполняется путем

преобразования чертежа. Преобразование чертежа включает построение связей между элементами по графическому изображению, выполнение анализа схемы, исходя из задачи и режима, выделение существенных элементов и упрощение схемы для передачи данных в модуль математического моделирования режима.

Одной из основных задач является формирование данных о математической модели схемы и последующей передаче их внешней программе решения. Сложное графическое представление схемы преобразуется под требования заданной задачи. Преобразование включает несколько этапов. Вначале отбрасываются отключенные участки схемы (те элементы, которые не имеют соединения с источником) и проверяется работоспособность всех оставшихся элементов.

Полученный участок схемы упрощается, т.е. исключаются избыточные соединительные линии и элементы, не используемые при моделировании выбранной задачи (например, средства измерения, защиты от перенапряжений и т.д). Данные сохраняются во внешнем файле базы данных, например Access (MDB).

В результате преобразования формируется таблица следующего содержания: название элемента (уникальное в пределах схемы); название порта связи элемента; название связанного с ним элемента; название порта связанного с ним элемента. Одна запись таблицы определяет одну одностороннюю связь для одного элемента. Чтобы получить все связи для одного элемента необходимо выбрать все записи с одинаковым названием.

Для работы с таблицами использован язык запросов SQL, который обеспечивает не только высокую гибкость в работе с таблицами данных, но и единость данных всего программного комплекса, включающего графическую систему проектирования, управления и математического моделирования.

Вызов внешней функции для решения задачи выполняется на использование OLE автоматизации, что позволяет легко добавлять и настраивать внешние программы, что отвечает требованиям к открытой архитектуре программной системы. Сам процесс решения рассматривается как некоторый «черный ящик», который получает данные и возвращает результат в качестве вызова внешней функции.

В качестве примера рассмотрим реализацию некоторых процедур программного обеспечения метода, которые используются в компоненте интерфейса для создания графа сети.

```
Процедура      УстановитьСвязи()
    базаДанных := ОткрытьБазуДанных()
    ЭлементыСхемы := ПолучитьВсеЭлементы();
    N := Количество(ЭлементыСхемы);
    // Просматриваем все элементы, кроме последнего
    Для i := 1 до N-1 повторять
        Элемент := ЭлементыСхемы[i]
        Точки := []; // Очистка массива точек соединения элемента
        Если Элемент.Тип == «Точка» то
            Точки.Добавить(Элемент.Координата, «T1»);
        Иначе Если Элемент.Тип == «ПолиЛиния» то
            // Линия имеет две точки соединения
            Точки.Добавить(Элемент.НачальнаяТочка, «T1»);
            Точки.Добавить(Элемент.КонечнаяТочка, «T2»);
        Иначе Если Элемент.Тип == «Блок» то
            // Блок имеет минимум одну точку соединения -
            // его точку вставки
            Точки.Добавить(Элемент.ТочкаВставки);
            // поиск в базе данных портов соединений
            // для этого элемента
            запрос := «SELECT p_name, p_x, p_y FROM ports
                WHERE p_item='Элемент.Имя'»;
            РезСтроки := базаДанных.Выполнить(запрос)
            // Найденные порты добавляются к точкам
            ДляКаждого РезСтроки как
                (Имя, X, Y)Точки.Добавить(X, Y, Имя);
            КонецЦикла
        КонецЕсли
        // теперь каждая точка проверяется на соединение со всеми
        // последующими элементами
        Для j:= i+1 до N повторять
            Элемент1 := ЭлементыСхемы[j];
            // Мультилиния не имеет точек и
```

```
                    // обрабатывается отдельно
                    Если Элемент.Тип == «Мультилиния» и
                            МультиСоединение(Элемент, Элемент1) то
                            Связь(Элемент, Элемент1)
                            // для всех остальных элементов проверяется на соединение
                            // каждая его точка.
                    Иначе
            ДляКаждого Точки (массив, временный с параметрами: название, координаты) как
             Точка (число итераций цикла определяется количеством точек элемента: генератор, трансформатор
двухобмоточный., трехобмоточный и т.д)
                    Если точка1=ЕстьСоединение(Элемент1,Точка) то
                        Связь(Элемент, Элемент1, Точка, точка1)
                                    Добавить в таблицу дуги две связи
                                    Связь1(Элемент, Элемент1, Точка, точка1)
                                    Связь2(Элемент1,Элемент, точка1, Точка)
                                КонецЕсли
                            КонецЦикла
                            КонецЕсли
                        КонецЦикла
                КонецЦикла
                ЗакрытьБазуДанных()
КонецПроцедуры
```

Выводы:

1. Разработана концепция автоматического создания динамической модели локального объекта электроэнергетической системы для текущего его состояния при учете санкционированных (управляющих) и несанкционированных внутренних и внешних воздействий. Основой для адаптации модели являются представления об объекте управления, которыми обладает квалифицированный персонал.

2. Основными принципами организации внеязыковой технологии программирования являются: детализация представления объекта в зависимости от уровня и иерархии управления, класса и вида технологической задачи; использование объектов (простых, сложных) с параметрической настройкой в отличие от объектно-ориентированного программирования; настройка объекта на основе графического изображения, позиций меню и опций, ситуаций (на основе информации от УТМ, результатов в ходе моделирования, информации, вводимой с клавиатуры); использование модели в виде множеств параметрических объектов взаимосвязанных в графической и символьной частях БД и БЗ, графов с последующей обработкой для создания динамических адаптируемых моделей технологических задач.

3. Создан метод, реализующий в различной степени разработанные принципы. Метод позволяет реализовать систему поддержки решений путем интеграции фаз интеллектуальной оперативной деятельности: контроля и идентификация состояния объекта управления, диагностики неисправности, планирования действий, оценки и выбора альтернативы. Эти фазы деятельности органично входят друг в друга, хотя у каждой свои характерные отличия, общими же являются знания по объекту управления.

4. В условиях тенденции усложнения управлением локальными объектами, диктуемой экономическими соображениями энергорынка, использование метода повышает эффективность системы автоматизированного управления.

ЛИТЕРАТУРА

1. Заболотный И.П., Ларин А.М., Заболотный Д.И. Автоматизированное рабочее место инженера предприятия электрических сетей // Энергетика и электрификация.- 1994.- №1.- С. 23-25.

2. Заболотный И.П. Развитие научных основ автоматизированных систем оперативного управления в энергетике // Сборник научных трудов ДонГТУ. Серия: Электротехника и энергетика, выпуск 2, Донецк:ДонГТУ.- 1998. – С. 189-193.

3. Заболотный И.П., Павлюков В.А. Применение компьютерных технологий для управления электрическими системами // Технічна електродинаміка, спеціальний випуск. К.- 1998. – С.90-99.

4. Акофф Р., Эмери Ф. О целеустремленных алгоритмах. – М.: Советское радио, 1974. - 84 с..

5. Кириленко А.В., Буткевич А.Ф., Павловский В.В. Экспертные процедуры диагностирования при оперативном управлении электрическими сетями в аварийных ситуациях // Техн. Электродинамика.-1995.-№1.- С. 66-73.

6. Новиков Ф.А. Дискретная математика для программистов. - СПб.: Питер, 2001. - 304 с.

7. Кристфидес. Н. Теория графов. Алгоритмический подход: пер. с англ. - М.: Мир, 1978. - 476 с.