НЕПРОЦЕДУРНЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ – ТЕОРЕТИЧЕСКАЯ И ПРАКТИЧЕСКАЯ ОСНОВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «САРГОН»

В.А. МЕНДЕЛЕВИЧ (ООО “НТЦ НВТ–Автоматика”)

В декабре 2016 г. исполняется 25 лет с момента выхода первой версии программно-технического комплекса “САРГОН”, который стал одним из самых эффективных отечественных средств автоматизации технологических процессов. “Круглой” дате посвящается цикл статей, раскрывающих важнейшие характеристики и механизмы построения современного ПТК.

Первая статья цикла описывает теоретические основы программно-технического комплекса “САРГОН”, определившие его облик и основные характеристики.

Ключевые слова: *САРГОН, АСУ ТП, непроцедурный язык, язык программирования, ПТК.*

1. ЦЕЛИ и ПРЕИМУЩЕСТВА ПРИМЕНЕНИЯ НЕПРОЦЕДУРНОГО ЯЗЫКА ДЛЯ АСУ ТП

В процессе разработки сложной системы участвует большое количество людей разных профессий, работающих в различных организациях. Нормативные документы, регламентирующие их взаимодействие, не охватывают значительной части вопросов. Многие из них сильно устарели (например, ГОСТы по АСУ ТП), не соответствуют современному уровню средств и технологий автоматизации и могут применяться лишь избирательно. Недостаточное взаимопонимание участников разработки становится большой проблемой.

Значительные сложности при разработке сложной системы возникают также из-за различий естественного человеческого мышления и механизма машинной обработки информации: естественное мышление преимущественно непроцедурно, оно основано на ассоциациях и связях. Чёткая последовательность действий, необходимая для работы вычислительной машины, требует от разработчика трансформации своего непроцедурного мышления в жёсткое процедурное описание. Такой режим “ручной компиляции” является одним из основных источников серьёзных ошибок. Эти ошибки, к тому же, трудно обнаружимы, так как они возникают на этапе постановки задач.

С ростом сложности создаваемых систем важность указанных проблем начала значительно возрастать. Для их решения в конце XX века сформировалось новое направление системного программирования – непроцедурные языки программирования.

***Непроцедурный язык** – это мощное средство разработки сложных программных систем, позволяющее сконцентрировать внимание разработчика на описании целей и правил, а не на последовательности действий по их реализации (т.е. описывается “что делать” вместо “как делать”).*

Непроцедурный язык ориентирован на конкретную предметную область (например, на автоматизацию технологических процессов на производстве). Различают два классических уровня реализации непроцедурного языка:

1. Формализованное описание предметной области и логики функционирования системы без генерации исполнительного кода. Такой непроцедурный язык называется языком спецификаций. Он используется как средство формализации задачи и её постановки программистам.
2. Полноценная управляющая система, построенная на непроцедурном языке. Сложности процедурной реализации перекладываются при этом на исполнительную систему, что обеспечивает ускорение разработки и значительное повышение надёжности создаваемых систем.

Уровни 1 и 2 часто становятся этапами создания непроцедурного языка. Так, в частности, произошло и с языком “САРГОН” [1, 2].

2. САРГОН НЕПРОЦЕДУРНЫЙ ЯЗЫК ДЛЯ АСУ ТП

Для программного обеспечения ПТК "САРГОН" непроцедурный язык программирования стал теоретической, а затем и практической, основой. Точнее сказать, первая версия ПО "САРГОН" была разработана в 1991 г. для решения практической задачи создания АСУ ТП высокого уровня сложности. В процессе разработки, при решении возникавших проблем, последовательно было применено несколько современных технологий системного программирования, которые сформировали в совокупности облик непроцедурного языка [3].

Классифицировать придуманное как непроцедурный язык помог автору через год Задыхайло Игорь Борисович – его научный руководитель с кафедры Системного программирования факультета Вычислительной математики и кибернетики МГУ, где автор завершал в это время второе образование. Это дало значительный импульс дальнейшему развитию системы, так как позволило использовать теоретическую базу целого направления системного программирования [4, 5].

Язык "САРГОН" решает следующие задачи по разработке системы:

1. Классификация элементов автоматизированного технологического комплекса (АТК), включающего в себя автоматизируемый объект и АСУ ТП.
2. Перечисление и описание основных характеристик элементов АТК.
3. Описание связей между элементами АТК.
4. Описание внутренней логики функционирования типовых элементов АТК.

После завершения этапа разработки производится генерация согласованных между собой комплектов документации, исполнительных кодов программ и конфигураций для всех вычислительных узлов системы.

Основами непроцедурного языка "САРГОН" стали:

- **Объектный подход** с представлением автоматизированного технологического комплекса как совокупности типовых программных моделей.
- **Использование математической модели конечного автомата** как основы для построения типовой программной модели.
- **Единая реляционная база** данных системы, в которой аккумулируются все описания и из которой генерируются документация, исполняемый код и конфигурации.
- **Сквозное проектирование** – поддержка всех этапов жизненного цикла и всех участников процесса создания системы.

Объектный подход стал в последнее время общепризнанной технологией создания сложных систем, поэтому повторять его преимущества не будем. В "САРГОН" объектный подход используется во всей полноте:

1. Каждому типу элемента АТК в языке "САРГОН" соответствует типовая программная модель, которая воспроизводит логику его функционирования. Типы объединены в иерархию классов с наследованием свойств.
2. Каждому элементу АТК в "САРГОН" соответствует экземпляр модели соответствующего типа, воспроизводящая его состояние и осуществляющая управление "своим" элементом АТК.
3. Организация взаимодействия программных моделей между собой и с внешней средой реализована на системном уровне и со стороны разработчика требует лишь описания связей между элементами АТК. При этом каждая модель функционирует в собственном потоке исполнения логически параллельно с другими моделями. Естественный параллелизм языка "САРГОН" избавляет пользователя от искусственного объединения в одно дерево выполнения слабо связанных процессов и обеспечивает технологически естественную декомпозицию сложной задачи управления на набор значительно более простых.
4. Обмен информацией между моделями производится как путём передачи данных с выходов одних на входы других моделей, так и путём передачи команд и сообщений между моделями в целом. Передача и исполнение команд осуществляется с учётом приоритета, блок приоритетов также реализован на системном уровне.

Математическая модель конечного автомата как основа для построения типовой программной модели изначально использовалась в "САРГОН" с целью повышения надёжности прикладного ПО. Такая основа обеспечивает:

- гарантированную детерминированность процесса управления и невозможность попадания системы в непредусмотренное программой состояние;

- единообразность процедур контроля и вычисления состояния объектов различного типа;
- единообразность выполнения операций по управлению объектами всех уровней: от клапана до технологической установки/энергоблока в целом;
- возможность эффективного управления системными ресурсами (каналами связи, базами данных и т.п.), представляемыми в языке как системные модели.

"Модельное" строение программ обеспечивает возможность полномасштабной проверки технологических программ управления любого объекта (от клапана до энергоблока включительно) на АРМ инженера АСУ ТП. Для проверки используется эффективная система отладки и моделирования из системы программирования "САРГОН", которая обрабатывает код, подготовленный для выполнения в реальном времени. В библиотеке системы программирования есть набор имитаторов типовых элементов, при подключении которых дополнительного программирования для моделирования не требуется.

Единая реляционная база данных – основа любой современной SCADA- и SoftLogic-программной системы – обеспечивает четкость и однозначность программного описания объекта и системы управления за счет нормализации структуры описания и уникальности ключа каждого компонента. Так как в ПТК "САРГОН" технология единой реляционной базы заложена в ядро системы, её преимущества полноценно использованы:

- SCADA (включая MES-компоненты) и SoftLogic-системы полностью интегрированы, по базе данных – любой компонент объекта или АСУ ТП описывается в базе данных однократно и затем используется во всех программных системах ПТК "САРГОН".
- Обеспечивается автоматическое отслеживание изменений, вносимых в описание любого компонента системы.
- Обеспечивается взаимное соответствие программного кода, документации и конфигураций ПО, так как они автоматически генерируются из единой базы данных.

Сквозное проектирование поддержка всех этапов жизненного цикла АСУ ТП и всех участников процесса создания системы. Этот принцип также был изначально заложен в языке "САРГОН" и чётко реализован во всех программных компонентах ПТК.

При этом непроцедурность языка "САРГОН" не превращается в фетиш – там, где естественным описанием является процедурное, например, в алгоритмах переключений, оно активно используется. Связывание различных форм описания в единое целое осуществляется автоматически на системном уровне, что является важным практическим достоинством языка "САРГОН".

3. ВАЖНЫЕ ОСОБЕННОСТИ ЯЗЫКА "САРГОН"

Прежде, чем рассматривать реализацию языка "САРГОН" в программно-техническом комплексе, отметим несколько важных особенностей, определяющих возможности языка.

Трёхуровневая иерархия Блок моделей/Класс/Тип

Как полноценная объектная система, "САРГОН" поддерживает многоуровневую иерархию классов с наследованием, инкапсуляцией, перегрузкой операций и другими свойствами, хорошо известными системным программистам. Однако, для большинства пользователей, пишущих технологические программы, иерархия сводится всего к трём уровням с чётким технологическим смыслом:

1. **Блок моделей** ("суперкласс") – объединяет все типовые программные модели общего технологического назначения. Например: блок моделей аналоговых технологических параметров, блок моделей дискретных технологических параметров, блок моделей арматуры, блок сложных алгоритмических моделей, блок моделей сетевых каналов связи, блок моделей контроллеров и т.д. Блок моделей в "САРГОН" имеет набор базовых свойств, который присущ всем входящим в него объектам.
2. **Класс моделей** – объединяет все типовые программные модели из конкретного блока с общими принципами функционирования. Например: класс запорной арматуры, класс регулирующей арматуры. Все объекты одного класса имеют свой общий набор свойств.
3. **Тип моделей** – объединяет все модели конкретного класса, имеющие одинаковые наборы параметров и методы их обработки. Например: задвижка с уплотнением на

закрытие, задвижка без уплотнения на закрытие, запорный клапан, свеча безопасности и т.д.

Такое трёхуровневое представление, являющееся частью непроцедурного языка "САРГОН", позволяет разработчикам АСУ ТП использовать все преимущества объектного подхода, не погружаясь в тонкости системного программирования.

Виды связей между программными моделями

В языке "САРГОН" предусмотрено два вида связей между программными моделями:

1. Между входным параметром одной модели (приёмника) и выходным параметром другой модели (источника) – классический способ связи, между компонентами прикладной программы, поддерживаемый во всех языках технологического программирования. В языке "САРГОН" он используется для передачи данных между программными моделями.
2. Между двумя программными моделями, находящимися в технологической связи мастер-подчинённый (например, горелка является мастером для установленного на ней клапане, а клапан, в свою очередь, мастером для ключа ручного управления собой). В "САРГОН" с помощью этого типа связи мастер получает информацию о состоянии подчинённых, передаёт подчинённым команды и контролирует допустимость выполнения команд, поступающих "со стороны" (в том числе, от оператора).

4. ПРОЕКТИРОВАНИЕ АСУ ТП НА ЯЗЫКЕ "САРГОН"

Из перечисленных в разделе 2 четырёх задач непроцедурного языка три относятся к процессу проектирования:

1. Классификация элементов автоматизированного технологического комплекса (АТК), включающего в себя автоматизируемый объект и АСУ ТП.
2. Перечисление и описание основных характеристик элементов АТК.
3. Описание связей между элементами АТК.

Для выполнения этих задач в состав ПО "САРГОН" входит система сквозного проектирования TkAconf. Она имеет следующие свойства:

- Вся информация о проекте АСУ ТП хранится в единой реляционной базе данных с высокой степенью нормализации.
- Поддерживается многопользовательский режим работы над проектом с возможностью одновременного внесения изменений с нескольких АРМ и автоматической блокировкой взаимно-недопустимых действий.
- Поддерживается мультипроектная разработка (ведение неограниченного количества проектов на одном сервере баз данных системы TkAconf) с автоматизированным межпроектным копированием связных блоков информации (проекта в целом, управления конкретным технологическим узлом, управления с конкретного контроллера и т.п.). Для каждого проекта система создаёт свою базу данных, предоставляя при этом широкие возможности одновременной работы с несколькими базами.
- Система содержит широкий набор справочников, объединённых в отдельную реляционную базу данных, информация из которой может использоваться во всех проектах АСУ ТП. Набор справочников имеет структуру, аналогичную типовому проекту, и содержит типовые решения для всех категорий компонентов АТК.

Среди выполняемых функций, в первую очередь, необходимо отметить следующие:

- контроль полноты, непротиворечивости и целостности информации о проекте;
- автоматическое отслеживание изменений, вносимых в элементы, во всех разделах проекта;
- автоматизация процесса проектирования, включая автоматическую генерацию согласованных конфигураций ПО всех вычислительных узлов АСУ ТП;
- выдача документации по проекту привязки ПТК, генерация связей с графической САПР в части КИПиА.

Система TkAconf имеет интуитивно-понятный современный интерфейс, удобный для пользователя, который обеспечивает различные способы представления информации:

- табличный с развитыми механизмами фильтрации, упорядочивания и группировки;
- графический в виде "дерева", в том числе, построенного пользователем;

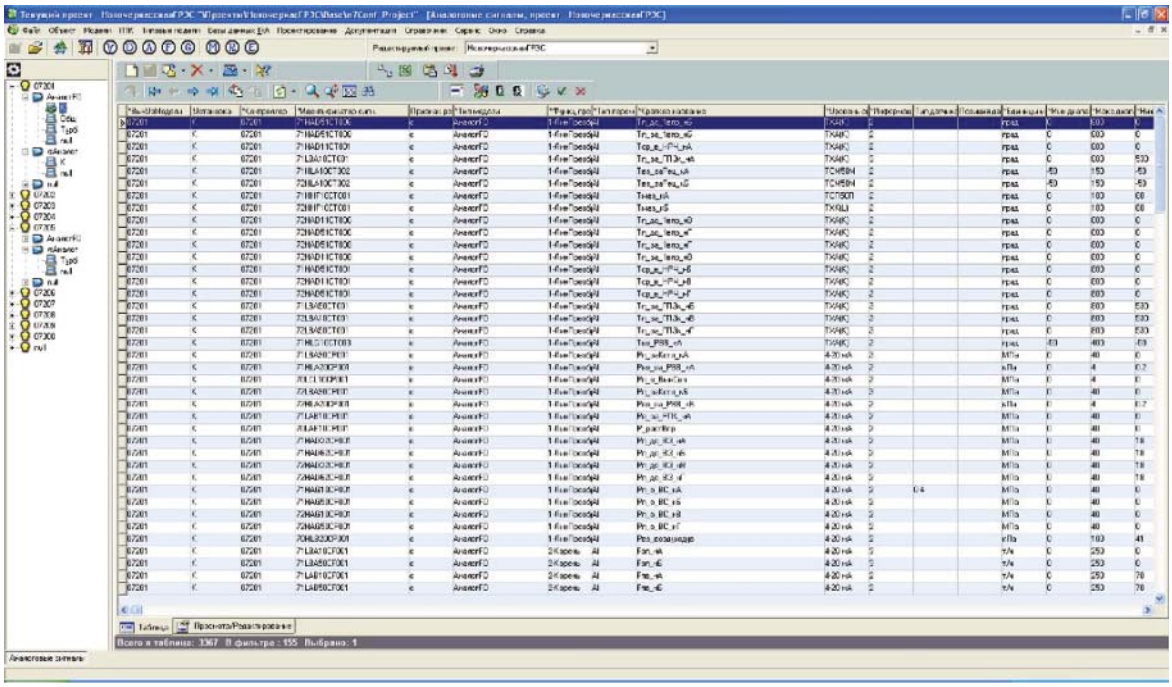

Рис. 1. Таблично-графическое представление перечня аналоговых параметров в TkAconf

- специализированной формы редактирования конкретного типа компонента АТК;
- их различных сочетаний (рис. 1).

Состав проекта:

- Описание объекта автоматизации и его элементов: структура объекта, исполнительные устройства, датчики технологических параметров (аналоговых и дискретных) содержатся в разделе “Объект”.
- Описания средств автоматизации: вычислительных узлов, модулей ввода-вывода, каналов ввода-вывода, сетей, конфигурационных групп и т.д. – содержатся в разделе “ПТК”.
- Описания моделей (типовых и конкретных моделей, соответствующих объектам автоматизации — арматура, аналоговые, дискретные, алгоритмические) и групповых операций с моделями содержатся в разделах “Модели” и “Типовые модели”.
- Общими для всех проектов являются Справочные таблицы, которые представляют собой справочную базу данных и доступны в разделе “Справочник” каждого проекта.

Система проектирования АСУ ТП на языке “САРГОН” поддерживает технологию **“Сквозное проектирование”**:

1. Работу над проектом в среде TkAconf начинают сразу после принятия решения о создании новой АСУ ТП.
2. Работу выполняют в следующем порядке:
   - Регистрация и настройка проекта.
   - Анализ объекта автоматизации.
   - Разработка и согласование технического задания.
   - Разработка и выполнение рабочего проекта.
   - Формирование и тестирование конфигурационных файлов.
   - Формирование проектной документации.

Результатами работы системы конфигурирования TkAconf являются:

- база данных проекта, содержащая полную информацию об объекте и системе;
- документация (содержательные приложения к документам ТЗ, ОТР, ИО, ПО);
- конфигурации для исполнительных систем реального времени всех контроллеров, АРМ и серверов АСУ ТП.

Таким образом, реализация непроцедурного языка “САРГОН” в системе сквозного проектирования TkAconf является эффективным средством описания элементов объекта автоматизации и АСУ ТП, связей между ними, процесса автоматизированного проектирования.

5. ПРОГРАММИРОВАНИЕ АСУ ТП НА ЯЗЫКЕ “САРГОН”

На долю системы непроцедурного программирования ложится самая сложная из

перечисленных в разделе 2 задач непроцедурного языка – описание логики функционирования типовых элементов АТК. Несмотря на объектно-компонентную декомпозицию объекта и АСУ ТП, на порядок снизившую сложность её решения, она остаётся наиболее трудоёмкой как на этапе разработки, так и наладки.

Система программирования TkAprog разработана для решения этой задачи непроцедурного языка "САРГОН". TkAprog обеспечивает составление высоконадёжных технологических программ и их эффективную отладку.

Система технологического программирования выполняет следующие функции:

- создание и редактирование типовых алгоритмов автоматизации любой сложности;
- создание и редактирование библиотеки типовых алгоритмов;
- автоматическая компиляция типового программного модуля с получением объектного файла для управления объектом автоматизации в режиме реального времени;
- контроль непротиворечивости и целостности информации обо всех типовых программных моделях проекта;
- отладка типовой программной модели в режиме имитации и в режиме реального управления на объекте автоматизации;
- автоматическое отслеживание изменений, вносимых в типовые модели и компоненты, во всех разделах проекта;
- выдача документации по проекту.

Система технологического программирования на непроцедурном языке обеспечивает:

- Представление сложного алгоритма управления типовым объектом в виде набора простых и интуитивно-понятных форм (табличных, графических, текстовых).
- Раздельное описание нормального хода процесса и действий в исключительных ситуациях (в соответствии с логикой технолога, зафиксированной в любой инструкции).
- Реализацию на системном уровне механизма диспетчеризации параллельно выполняющихся процессов (разработчику в большинстве случаев можно не задумываться об организации программного цикла).
- Реализацию на системном уровне блока приоритетов для передачи и выполнения команд с возможностью явного указания приоритетов команд, запретов, блокировок.
- Проверку непротиворечивости независимо составленных описаний компонентов объекта и системы.
- Возможность имитационного выполнения технологических программ для их отладки на полигоне.

В "САРГОНе" реализованы три уровня свободы программирования типовой модели разработчиком АСУ ТП. Все типовые модели, относящиеся к одному блоку моделей, имеют одинаковый уровень свободы программирования:

1. Логика работы модели настраивается выбором вариантов из меню и значениями коэффициентов. Изменение типовой логики разработчиком АСУ ТП невозможно.
2. Логика работы модели задаётся таблицей состояния модели. Разработчик АСУ ТП может составить и изменить типовую логику работы модели в фиксированном формате описания.
3. Логика работы модели полностью задаётся разработчиком АСУ ТП. Язык содержит удобные формы составления программ и типовые библиотечные элементы, но разработчик имеет возможность дополнить их произвольным кодом на стандартных языках ST и FBD.

Рассмотрим подробнее особенности каждого уровня и их назначение:

1. Главными достоинствами моделей уровня программирования 1 являются простота и эффективность работы в режиме реального времени. Логика функционирования модели задана в самой виртуальной машине "САРГОН" (в коде на языке C++). Разработчик АСУ ТП имеет возможность выбрать подходящий класс модели, один из поддерживаемых выбранным классом типов, изменять значения констант (в системе проектирования) и коэффициентов (в системе реального времени). При этом трёхуровневая иерархия "блок моделей/класс/тип", описанная в разделе 3, в дополнение к эффективности работы обеспечивает сочетание простоты использования типовых моделей в проекте (привязка в большинстве случаев осуществляется автоматическими процедурами системы проектирования) и гибкости. Например, для моделей аналоговых технологических параметров: алгоритм функции сигнализации о превышении значением параметра уставки описан в классе модели, уровень сигнализации (предупредительная или аварийная) и текст сообщения – в типе модели, а величина уставки – коэффициентом модели конкретного аналогового параме-

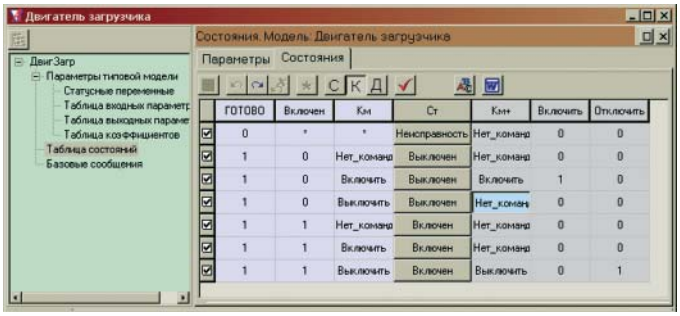

Рис. 2. Таблица состояний типовой модели арматуры "Двигатель загрузчика"

тра. Эффективность и простота применения логики этого уровня обеспечили ему активное использование – модели уровня 1 составляют большинство блоков моделей: моделей дискретных технологических параметров, моделей аналоговых технологических параметров, всех блоков системных моделей (каналов связи, баз данных, модулей УСО и т.п.). В общем числе программных моделей АСУ ТП модели уровня 1 также преобладают и составляют от 70 до 90 % в разных системах.

2. Логика функционирования типовой модели уровня программирования 2 задаётся таблицей состояния модели. Разработчик АСУ ТП имеет возможность в наглядной табличной форме и с высокой надёжностью описать алгоритм управления среднего уровня сложности (рис. 2) – в левых столбцах таблицы перечислены все возможные комбинации критериев состояния модели, а в правых – выполняемые в идентифицированном состоянии действия. Такие типовые модели по эффективности работы в реальном времени близки к моделям уровня 1, но сложность алгоритмов, кодируемых автоматной таблицей, ограничена – описание в такой форме алгоритма высокой сложности получается трудоёмким и трудно воспринимаемым. В языке "САРГОН" на этом уровне свободы описываются алгоритмы всего одного, но очень важного блока моделей – арматуры. Использованная форма описания логики работы гарантирует по построению, что арматура не может оказаться в не предусмотренном технологической программой состоянии. Так как класс арматуры является базовым для реализации функции дистанционного управления и отработки технологических защит, гарантированная надёжность является важным свойством типовых моделей уровня программирования 2. В различных проектах доля программных моделей уровня 2 составляет от 0 (в некоторых информационных системах) до 30 %.
3. Все типовые модели с уровнем свободы программирования 3 собраны в одном блоке моделей – "Алгоритмические модели". Технологическая программа для типовой модели этого блока на языке "САРГОН" представляет собой совокупность форм заданной структуры (*секций*):
   - *Таблица параметров* – описывает данные, обрабатываемые моделью и ее взаимодействие по данным с другими моделями. Алгоритмы модели могут использовать только те параметры, которые описаны в таблице. Такая локализация обеспечивает независимость алгоритмов от внешнего окружения (привязка формальных параметров к фактическому размещению в контроллере выполняется автоматизированной процедурой системы проектирования).
   - *Таблица исполнителей команд* – описывает все модели, на которые данная модель посылает или запрещает команды. Выдавать команды и налагать запреты можно только на модели, указанные в данной таблице. Это обеспечивает возможность построения сложных многокомпонентных программ, не привязанных к конкретной аппаратной реализации. Привязка также осуществляется автоматизировано в системе проектирования.
   - *Таблица перечислимых типов* – содержит описание всех значений параметров перечислимого типа, используемых в данной типовой модели. Например, параметр перечислимого типа "Состояние задвижки" принимает значения: "Открыта", "Закрыта", "Неисправна", "Открывается", "Закрывается" и т.п. Именно к таким переменным относятся в языке "САРГОН" все состояния и команды. Использование параметров перечислимого типа намного более естественно для человека, чем машиноориентированных "0/1".
   - *Таблица состояний* – описывает идентификацию состояния модели по значениям её параметров. Использование математической модели конечного автомата как основы построения этой таблицы

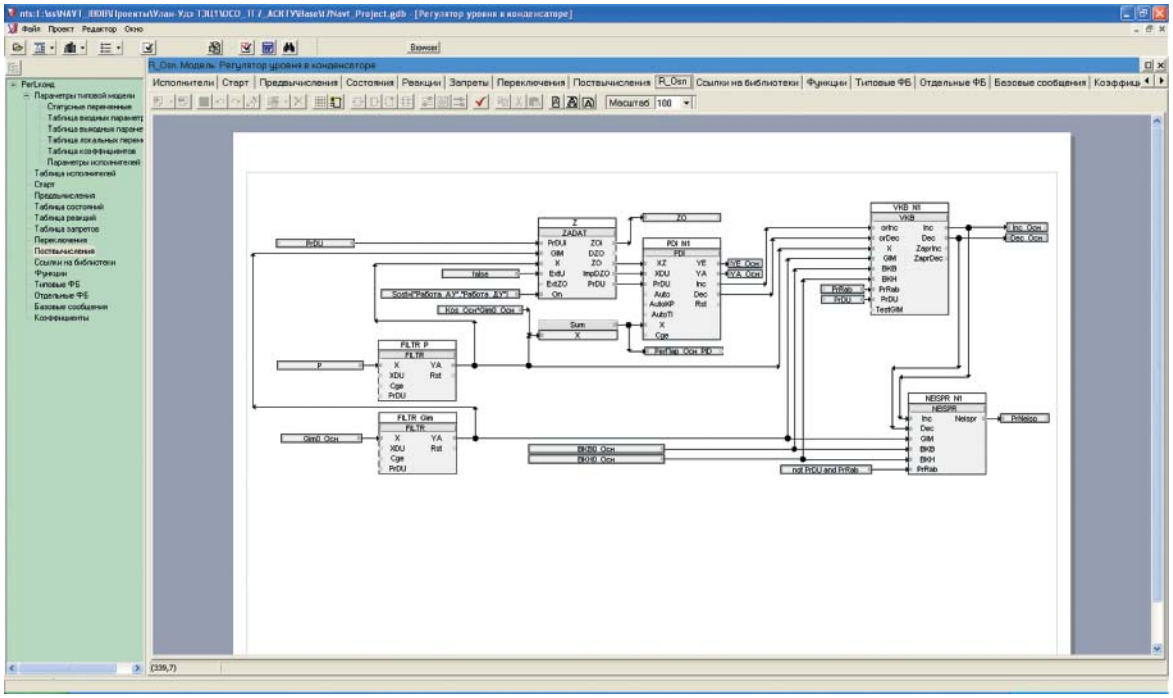

Рис. 3. Регулятор уровня конденсата в конденсаторе

гарантирует непротиворечивость и наглядность описания функции вычисления состояния модели, что обеспечивает наблюдаемость и управляемость модели в любом режиме её функционирования. В отличие от моделей с уровнем программирования 2, исполнительной части эта таблица не содержит.

- *Таблица реакций* – в наглядной табличной форме описывает все существенные для модели события, и ответные реакции модели на эти события. В том числе, описываются все нештатные ситуации. Реакция или осуществляет одномоментный блок действий (одношаговая реакция), или запускает многошаговое *Переключение*. Все реакции выполняются параллельно.
- *Таблица запретов* – описывает запреты на передачу команды подчиненным. Виртуальная машина “С*А*РГОН” обеспечивает автоматическое отслеживание таблично описанных запретов с учётом всех факторов: запрещаемые команды могут исходить от оператора, от физических органов управления, от других программных моделей; на одно действие одновременно может быть наложено несколько запретов с разными приоритетами, которые снимаются и устанавливаются в разной последовательности и т.д. Результат действия запрета (команда проходит подчинённому, или отбрасывается) определяется соотношением приоритетов запрета и поступившей команды.
- *Таблицы переключений* – описывают все переключения модели (последовательности действий, приводящие к изменению состояния модели). Переключения могут быть инициированы или внешней командой (от оператора, от другой программной модели, от ключа управления), или Реакцией. Переключения состоят из последовательности шагов, каждый из которых имеет условие начала, блок выполняемых на шаге действий и условие окончания.
- *Вычислительные блоки* – свободно программируемые части технологической программы на языке “С*А*РГОН”. Блоки пишутся на языках ST и FBD из стандарта МЭК 61131-3. FBD-диаграмма для TkAprog является стандартным вызовом функции, поэтому она может быть вставлена в любое место программы, написанной на ST. Классическим примером применения FBD является описание работы контура регулирования (рис. 3).

Свободная форма программирования секций вычислительных блоков “С*А*РГОН” (их в описании модели три: *Старт* – выполняется однократно при первом вызове модели, *Предвычислений* – выполняется в каждом программном цикле до всех остальных секций модели и *Поствычислений* – выполняется последней в программном цикле) позволяет разработчику АСУ ТП описать всю логику работы модели, не используя особенности непроцедурного языка. В этом случае программа деградирует до стандартной процедурной на МЭК 61131-3 с расширением на возможности объектного подхода. Но научившиеся программировать на языке “С*А*РГОН” никогда на это не согласятся!

Разделение описания логики работы модели на указанные формы обеспечивают максимальную ясность технологического смысла каждого выражения и перекладывают на системный уровень организацию взаимодействия между ними.

В правильно спроектированной АСУ ТП на языке “С*А*РГОН” доля алгоритмических моделей невелика и редко превышает 10 %, но их обработка часто занимает больше 50 % времени программного цикла контроллера.

6. ЕЩЁ РАЗ О НАДЁЖНОСТИ АСУ ТП, РАЗРАБОТАННОЙ НА ЯЗЫКЕ “С*А*РГОН”

Язык непроцедурного программирования “С*А*РГОН” создавался для применения в АСУ ТП особо опасных объектов, поэтому надёжность создаваемых систем была одним из главных приоритетов при разработке языка и его реализаций.

Кроме упоминавшихся выше средств повышения надёжности: наглядности и естественности описаний, возможности доказательства правильности программ, реализации трудоёмких операций на системном уровне используется множество других средств, повышающих надёжность систем на “С*А*РГОН”. Например:

- **Многозначная логика.** Во-первых, в языке предусмотрена универсальная обработка недостоверных данных. Так как контроль достоверности данных, используемых для принятия ответственных решений, составляет значительную часть технологической программы в АСУ ТП опасных объектов, то системная реализация логики с учётом достоверности позволяет значительно упростить программы и повысить их надёжность. Во-вторых, возможность использования многозначных переменных перечислимого типа вместо двузначных 0/1 также позволяет сократить сложность и повысить надёжность многих программ.
- **Корректность программ на языке ST по построению.** Одной из наибольших угроз надёжности разрабатываемых технологических программ являются тривиальные ошибки разработчика АСУ ТП в процессе написания текста программы (несогласованное открытие/закрытие простых и операторных скобок, ошибочное написание идентификаторов переменных, бесконечные циклы и т.п.). Примеры тяжёлых последствий таких ошибок оказали существенное влияние на весь ход развития языков программирования. В языке “С*А*РГОН” для максимального сокращения числа подобных ошибок реализован редактор языка ST, обеспечивающий корректность программ “по построению”:
  - выбор параметров производится из списка доступных и допустимых по типу переменных;
  - выбор операций из перечня допустимых для указанных операндов;
  - операторы управления вставляются согласованными блоками (например, IF-THEN-ELSE) и удаляются только целиком;
  - бесконечные циклы запрещены;
  - глубина вложенности циклов ограничена и т.д.

7. ИСПОЛНЕНИЕ ПРОГРАММ НА НЕПРОЦЕДУРНОМ ЯЗЫКЕ “С*А*РГОН”

Исполнение технологических программ, написанных на непроцедурном языке “С*А*РГОН”, реализовано через виртуальную машину языка, которая устанавливается на все компьютеры и контроллеры оперативного контура АСУ ТП и обеспечивает:

- Загрузку и исполнение программ, написанных на непроцедурном языке, в контроллерах и на АРМ операторов.
- Высокую эффективность выполнения программ в реальном времени (используется генерация промежуточного кода с Just-In-Time компиляцией, или прямая компиляция в объектный код; эффективность исполнения кода соответствует C++ и значительно выше, чем у программ, использующих любые формы OLE (в том числе OPC)).

- Переносимость прикладных программ на вычислительные платформы различного типа как по ОС, так и по типу процессорного ядра (виртуальная машина уже трижды переносилась на новые ОС и процессорные платформы).
- Независимость текста прикладных программ от типов и конфигурации аппаратных средств.
- Отладку технологических программ в терминах и формах непроцедурного языка.
- Имитационную загрузку и отладку программ из всех контроллеров оперативного контура на одном АРМ инженера АСУ ТП.
- Удалённую отладку программ на контроллерах (пошаговую и в реальном времени).

При использовании языка “САРГОН” для создания распределённых систем управления важна способность виртуальной машины языка осуществлять динамическое связывание моделей, расположенных на разных узлах сети:

- Виртуальная машина при загрузке обнаруживает потребность в данных с других контроллеров и автоматически конфигурирует соответствующие каналы и кадры обмена.
- В системе реального времени активно используется динамическое связывание между программными моделями при загрузке программ по идентификаторам (уникальным именам). Это позволяет минимизировать описание связей при разработке прикладного ПО.

Большие преимущества перед традиционными даёт использование непроцедурного языка в процессе наладки АСУ ТП:

- Модельный характер компонентов системы максимально облегчает процесс имитационного моделирования работы АСУ ТП на АРМ разработчика и на полигоне.
- Многоуровневая декомпозиция объекта и системы делает каждый её элемент легко контролируемым и управляемым.
- Таблично-диаграммная форма большинства описаний технологической программы обеспечивает наглядность представления информации (рис. 4).

8. АЛЬТЕРНАТИВНЫЕ ПРИМЕРЫ НЕПРОЦЕДУРНЫХ ЯЗЫКОВ ДЛЯ СОЗДАНИЯ АСУ ТП

Непроцедурные языки как языки спецификаций начали использоваться ещё в 60-х годах XX века. Исполняемые непроцедурные языки появились в конце 80-х годов XX века [5].

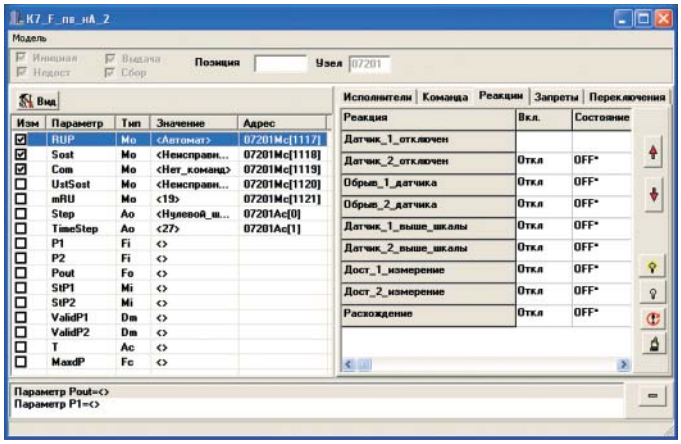

Рис. 4. Отладочное окно аналогового технологического параметра

Однако широкого распространения в АСУ ТП они не получили, несмотря на серьёзные преимущества, обеспечиваемые их применением. До недавнего времени, кроме “САРГОНа”, автору были известны примеры применения непроцедурных языков для АСУ ТП только в аэрокосмической и военной сфере (в том числе, в ПО авионики Airbus).

В стандартах Газпрома для описания логики работы АСУ ТП используются “причинно-следственные таблицы”, которые являются локальным языком спецификаций.

В последнее время существенно пробудился интерес к непроцедурным языкам в Росатоме. После аварии на АЭС “Фукусима” МАГАТЭ ужесточила требования к безопасности атомных станций, что привело к пониманию необходимости большей формализации процесса разработки алгоритмов и их перевода в исполняемый код. Следствием этого понимания стал запуск в нескольких институтах Росатома процесса разработки собственных непроцедурных языков. Пока эти процессы находятся в начальной стадии.

Автор и коллектив “НВТ-Автоматика” готовы способствовать разработкам коллег для применения непроцедурных языков в АСУ ТП в новых отраслях промышленности.

9. ЗАКЛЮЧЕНИЕ

Применение непроцедурных языков программирования – одно из наиболее перспективных направлений развития АСУ ТП. В последние годы, в связи с ростом требований к надёжности систем управления, разработка непроцедурных языков значительно активизировалась, но часто производится разработ-

чиками неосознанно, что заставляет их “изобретать велосипед”.

В ПТК “С*А*РГОН” непроцедурный язык программирования заложен в основе системы, успешно применяется и развивается в течение 25 лет. Его использование обеспечило следующие свойства ПО “С*А*РГОН”:

- Гарантированная надёжность технологических программ.
- Эффективное средство формализации описаний.
- Поддержка всех уровней и всех этапов жизненного цикла системы управления.
- Мультипроектная поддержка с эффективным тиражированием.
- Встроенная поддержка имитационного моделирования.
- Независимость прикладных программ от конфигурации ПТК.

Использование передовой технологии – непроцедурного языка программирования – одна из важных составляющих высокой надёжности АСУ ТП на базе ПТК “С*А*РГОН”, отмечаемой нашими заказчиками.

Список литературы

1. *Менделевич В.А.* Диссертационная работа на соискание ученой степени кандидата физико-математических наук “Система сквозного проектирования автоматизированных систем управления промышленными объектами на примере энергетических станций”, МГУ им. Ломоносова – Москва, 1997.
2. *Менделевич В.А.* Программные средства ПТК “С*А*РГОН 6.5” – Автоматизация и IT в энергетике, № 6, 2010.
3. *Менделевич В.А.* Непроцедурные языки – новое поколение средств разработки АСУ ТП – Промышленные АСУ и контроллеры № 1, 2000.
4. *Агафонов В.Н.* Спецификации программ: понятийные средства и их организация. – Новосибирск: Наука, 1990.
5. *Андрианов А.Н., Ефимкин К.Н., Задыхайло И.Б.* Непроцедурный язык Норма и методы его реализации // 1990. Сборник Языки и параллельные ЭВМ. С. 1-37.

Менделевич Владимир Анатольевич *– канд. физ.-мат. наук, генеральный директор ООО “НТЦ НВТ-Автоматика”.*
Телефон (495) 361-23-34. E-mail: mail@nvtav.ru