

Методы автоматизации в среде СИНТАР

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ»

Описаны методы автоматизации ввода и генерации данных в среде разработки контроллерного ПО СИНТАР. Введена классификация методов автоматизации ввода данных. Кроме общепринятых методов, описаны такие нетрадиционные методы, как вложенные итерации для генерации многоуровневых структур, SQL-шаблоны программных модулей, автогенерация сигналов, сочетание типовых элементов с автогенерацией, локализация программ с заменой множества обратываемых сигналов.

Ключевые слова: САПР АСУТП СИНТАР, Softlogic, программируемый контроллер, программное обеспечение, сигнал, база данных, автоматизация ввода данных.

Введение

Автоматизация разработки программ имеет две приятные особенности: во-первых, она уменьшает объем «ручной» работы, и, во-вторых, работа, сделанная автоматически, не содержит ошибок (с точностью до остаточных ошибок средств автоматизации).

Семейство СИНТАР [1] - это инструментальные системы класса SoftLogic, ориентированные на разработку крупномасштабных АСУ промышленного применения. Эффективность подобных систем напрямую зависит от того, насколько они насыщены средствами автоматизации и какова «емкость» этих средств. Здесь мы предлагаем обзор методов автоматизации, используемых в средах семейства СИНТАР, с попыткой анализа и классификации. Помимо рекламных целей такой обзор может дать инструмент для сравнения систем разработки ПО АСУ и оценки их эффективности.

Рассмотрим группы методов автоматизации в порядке увеличения их емкости или масштаба автоматизируемых объектов.

Итерации

Итерация – это средство тиражирования объекта с заданным количеством повторений и изменением содержимого. Обычное копирование строки или фрагмента схемы – это пример самой примитивной итерации.

а) Простые итерации

Самый популярный пример простой итерации в программировании – это оператор **for**. В СИНТАРе итерации применяются также для автоматизированного построения различных структур и описаний.

Например, с помощью итерации можно заполнить нужными значениями столбец таблицы описаний сигналов. В этом случае задается постоянная часть значения, к которой приписывается изменяемый итератор. Указываются нижняя и верхняя границы итератора, шаг его изменения и количество повторений значения на каждом шаге.

б) Вложенные итерации

Вложенные итерации удобны для описания многоуровневых структур. Примером вложенной итерации в СИНТАРе является средство, применяемое для построения многоуровневой структуры оборудования АСУ. Здесь элемент структуры определяется именем и аппаратным адресом. Например, Reg12 – имя некоторого регистра, а 12 – его адрес внутри охватывающей структуры. Для описания структуры некоторого устройства достаточно задать несколько итераций, например:

Уровень	Префикс	Адрес
1	Port	A,B,C,D
2	Reg	0-15
3	Bit	0-7

чтобы получить 3-уровневую древовидную структуру, включающую 512 элементов.

Шаблоны

Параметризованные шаблоны используются во многих языках программирования. Это обычная макроподстановка. В текстовых языках СИНТАР они также используются, но наряду с ними широко применяются SQL-шаблоны.

SQL-шаблон – это шаблон, который кроме обычных параметров содержит SQL-запрос к базе данных. В результате его расширения для каждой строки результата запроса генерируется текст шаблона с подставленными вместо параметров значениями из базы данных. Если учесть, что вся информация о разрабатываемой АСУ (программы, сигналы, конфигурации) находится в базах данных, то понятно, насколько популярно и эффективно это средство автоматизации: с помощью одного SQL-шаблона можно сгенерировать сотни строк исходного кода.

Например, можно автоматически получить вызовы подпрограммы предварительной обработки (фильтрации, достоверизации) входных аналоговых сигналов для каждого сигнала, обрабатываемого данной стойкой. Этот вызов может включать все необходимые свойства сигнала, взятые из базы данных.

Другой метод использует шаблоны определенной структуры для описания имен и других свойств сигналов. Такие шаблоны используются для автоматической генерации описаний сигналов при вводе в базу данных сигналов.

Типовые структуры

Типизация – это наиболее распространенный метод автоматизации, используемый в системах разработки программ вообще и в средах СИНТАР – в частности. Она охватывает различные объекты, начиная от типов данных и заканчивая программно-аппаратными структурами произвольной глубины и сложности. Эффективность этого метода обеспечивается легкостью многократного использования заранее продуманных и проверенных алгоритмических и структурных решений.

а) Типовые элементы

Этот метод широко используется в средах СИНТАР. Собственно, поддерживаемый стандартный язык FBD [2] основан на применении типовых блоков. Любой программный элемент, текстовый либо графический, может использоваться многократно как в том проекте, где он описан, так и в любом другом программном проекте, причем использование предполагает не создание копии, а ссылку на единственный экземпляр элемента. Это снимает проблему отслеживания изменений типового элемента.

Типизация в средах СИНТАР распространяется и на аппаратуру. Аппаратная структура любого уровня может быть объявлена как типовая, после чего она может многократно вставляться в необходимые места дерева структуры конфигурации. Этот метод особенно характерен для описания структуры типовых устройств связи с объектом. Типовые структуры применяются также при описании паспортных данных элементов аппаратуры.

Важно отметить, что определение всех указанных типовых объектов, как программных, так и аппаратных, предоставляется пользователю. Это дает возможность накапливать библиотеки типовых элементов, характерных для данной прикладной области, и использовать их в будущих разработках.

б) Классы и типы данных

Классы представляются естественным продолжением типовых элементов: зародышем классов стало появление библиотек, в которых группы связанных функций использовали общие данные.

СИНТАР – это объектно-ориентированная система, поэтому она поддерживает пользовательские классы. Сочетание с визуальным программированием [3] дает дополнительную эффективность: пользователю не требуется знать все тонкости, связанные с классами и объектами. Например, он может даже не подозревать о существовании конструкторов и деструкторов – и объявления классов, и их инициализацию система генерирует автоматически на основе минимальных графических данных о полях и методах.

СИНТАР предлагает небольшой набор предопределенных типов данных, из которых пользователь может строить производные типы: структуры, массивы и строки, а также любые их комбинации.

Преимущества объектной парадигмы и типизации данных общеизвестны – и здесь мы не будем их повторять.

Автогенерация сигналов

Программы АСУ, по существу, являются программами обработки сигналов. Количество сигналов, «населяющих» крупномасштабную систему, достигает десятков и сотен тысяч. Описание каждого сигнала в виде набора свойств должно быть введено в базу данных, которая затем используется многими компонентами ПО АСУ как при разработке, так и в эксплуатации. Набор свойств в зависимости от типа сигнала колеблется от 4-5 до 20 - 30 числовых и строковых значений.

Сказанное объясняет, почему автоматизация описания сигналов имеет критическую важность. Кроме описанных выше методов итераций и копирования (от

отдельных сигналов до целых таблиц) СИНТАР обеспечивает автоматическую генерацию сигналов с использованием двух методов.

Первый метод базируется на том факте, что сигнал некоторых типов сопровождается несколькими «подчиненными» сигналами, которые связаны с ним настолько устойчиво, что существование подчиненного сигнала без главного может рассматриваться как нарушение ссылочной целостности базы данных. Другими словами, при вводе главного сигнала система должна автоматически ввести все его подчиненные сигналы, а при удалении – удалить подчиненные сигналы. Кроме того, некоторые свойства главного сигнала могут определять значения свойств подчиненных сигналов, что должно автоматически отслеживаться как при вводе главного сигнала, так и при изменении его свойств.

Пользователь описывает правила автогенерации при описании структуры типа сигнала. Эти правила базируются на принятой модели и включают общие условия генерации и удаления всех подчиненных сигналов, а также следующие данные для каждого подчиненного сигнала:

- тип сигнала;
- условия генерации/удаления;
- список «наследуемых» свойств (полей) главного сигнала;
- список полей, значения которых зависят от полей главного сигнала с указанием зависимостей.

Условия и выражения для зависимых полей записываются на упрощенном подмножестве SQL, в котором допустимы встроенные и пользовательские функции.

На основе описания автогенерации система создает на сервере базы данных триггеры, которые автоматически осуществляют необходимые действия при вставке, изменении или удалении главного сигнала.

Эффективность данного метода достаточно высока: например, для каждого входного аналогового сигнала генерируется до 25 подчиненных сигналов различных типов. При этом благодаря реализации с помощью сервера автогенерация не зависит от источника ввода главных сигналов: их может набирать пользователь с помощью редактора сигналов, они могут экспортироваться или копироваться из другой базы данных, а также генерироваться средствами сравнения и синхронизации баз данных (см. ниже).

Идея второго метода автоматической генерации сигналов основана на устойчивой связи между некоторыми (типовыми) программными модулями или методами классов и сигналами, связанными с их входами/выходами. Если учесть, что программный модуль связан с набором сигналов определенных типов во всех случаях его использования, то можно «намертво» связать установку на схему экземпляра этого модуля с генерацией экземпляра соответствующего набора сигналов и автоматической вставкой набора на схему программы и в базу данных сигналов, а удаление экземпляра модуля - с удалением набора сигналов.

Уникальные имена и другие выбранные пользователем свойства сигналов генерируются с учетом параметров набора, список которых также определяет пользователь. При установке программного модуля, связанного с набором сигналов, пользователю предъясняется диалог для определения конкретных значений параметров, после чего все дальнейшие действия происходят автоматически.

Эффективность данного метода зависит от количества типовых модулей, устойчиво связанных с определенными сигналами, что, в свою очередь, связано с

качеством проектирования типовых модулей и классов. На практике число таких модулей обычно не меньше 50% общего объема библиотеки.

Локализация программ

Наиболее значительный выигрыш получается за счет использования полной программы в разработке, для которой она не предусматривалась. Параметризованные шаблоны тут не проходят по двум причинам:

- практически никогда заранее не известно, что данная программа будет использоваться где-то еще;
- даже если бы это и было известно, то параметрами нужно было бы сделать все используемые сигналы, а их количество для одной стойки доходит до нескольких тысяч.

Для переноса программного проекта, разработанного для одной аппаратной среды, в другую в новой версии СИНТАР [4] предусмотрена специальная операция. По смыслу она похожа на локализацию, т.е. приспособление программы к местным условиям, поэтому использован этот же термин.

Локализация программного проекта заключается в автоматической настройке, при которой могут быть изменены имя проекта, имена программных разделов и глобальных переменных (сигналов). Диалог локализации позволяет изменить:

- имя проекта и имена разделов программы (при необходимости);
- имена глобальных переменных и сигналов (возможно, с применением шаблонов);
- все используемые БД на выбранные из списка БД новой разработки.

Неувязки, связанные с отсутствием в новых БД необходимых сигналов или с наличием неиспользуемых сигналов, разрешаются при открытии программного проекта в новой среде.

Вспомогательные средства

Приведенная классификация охватывает методы автоматизации ввода данных в систему разработки. За ее пределами остались такие важные средства, как поиск и замена, имеющиеся в каждом приличном редакторе. В редакторах графических и текстовых языков среды СИНТАР указанные средства также имеются, причем область их действия распространяется на весь проект, так что в этом смысле они тоже «приличные».

СИНТАР располагает средствами автоматического сравнения различных структур с возможностью управляемой пользователем синхронизации. Такие возможности реализованы для баз данных сигналов (сравнение структур БД или сигналов и их свойств) и программных проектов (сравнение элементов всех уровней, начиная с разделов программы и заканчивая комментариями).

Реализованы средства автоматической регистрации изменений и ведения журнала коррекций для программных проектов и баз данных сигналов.

СИНТАР 2007 способен автоматически строить многоуровневое множество схем конфигурации по описанию структуры конфигурации и связей между вершинами дерева. Это не относится к автоматизации ввода данных, но, тем не менее, представляет собой нетривиальное преобразование данных в визуальную форму.

Новая модификация среды СИНТАР поддерживает пользовательскую классификацию сигналов на уровне базы данных. Это позволяет облегчить выбор сигналов при проведении определенных групповых операций (например, импорта необходимых сигналов из базы данных в программный проект или отбора сигналов для размещения на сетевых линиях связи). Попробуйте выбрать из 2000 сигналов всего 400! Если же сигналы заранее классифицированы, Вы динамически составляете условие фильтрации, которое наряду с классами может включать значения любых свойств сигналов, и получаете требуемое подмножество сигналов с помощью SQL-запроса, который автоматически генерируется системой.

Выводы

Среда разработки ПО АСУ, как правило, содержит инструменты двух видов: редакторы и генераторы. Редакторы позволяют вводить информацию о разрабатываемом проекте и сохранять ее в базах данных, а генераторы – обрабатывать данные, накопленные в БД, и получать различного рода результаты. Наша классификация методов автоматизации затронула, в основном, средства ввода данных (редакторы), поскольку генераторы – это автоматы по определению. Не рассматривались также средства автоматизации отладки, которые описаны в работе [5].

Тем не менее, предложенная классификация охватывает стандартные методы (например, итерации, классы, типовые элементы), не очень известные и во все экзотические (SQL-шаблоны, автогенерация сигналов, сочетание типовых элементов с автогенерацией). Поэтому авторы считают, что эти методы и сама классификация представляют определенный интерес для специалистов в области АСУ.

Список литературы

1. САПР СИНТАР-3. Начало реализации концепции построения перспективных САПР систем сбора информации и управления технологическими процессами/ В.Г. Сухоребрый, А.С. Гристан, В.Г. Джулгаков, Д.В. Джулгаков // Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. Нац. аэрокосм. ун-та «ХАИ». – Вып. 21. - Х., 2004. - С. 83 – 91.
2. International standard 1131-3, Part 3: Programming languages, IEC, Division Automatismes Programmables, First edition, 1993.
3. Визуальное объектное программирование промышленных контроллеров / В.Г. Сухоребрый, А.С. Гристан, В.Г. Джулгаков, и др. // Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. Нац. аэрокосм. ун-та «ХАИ». – Вып. 24. - Х., 2004. - С. 224 - 230.
4. Сухоребрый В.Г. СИНТАР-3 и СИНТАР 2007: сравнительный анализ систем разработки контроллерного ПО / В.Г. Сухоребрый, А.С. Гристан, Д.В. Джулгаков // Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. Нац. аэрокосм. ун-та «ХАИ». – Вып. 38. - Х., 2008. - С. 170 - 176.
5. СИНТАР-3 Mono – САПР для одномашинной АСУ / В.Г. Сухоребрый, А.С. Гристан, В.Г. Джулгаков и др. // Открытые информационные и компьютер-

ные интегрированные технологии: сб. науч. тр. Нац. аэрокосм. ун-та «ХАИ». – Вып. 24. - Х., 2004. - С. 257 - 262.

Рецензент: д-р техн. наук, проф., зав. каф. А.Г. Гребенников, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков

Поступила в редакцию 12.11.09.

Методи автоматизації в середовищі СІНТАР

Описано методи автоматизації введення і генерації даних у середовищі розробки контролерного ПЗ СІНТАР. Введено класифікацію методів автоматизації вводу даних. Окрім загальноприйнятих методів описано такі нетрадиційні методи, як вкладені ітерації для генерації багаторівневих структур, SQL-шаблони програмних модулів, автогенерація сигналів, поєднання типових елементів з автогенерацією, локалізація програм із заміною набору оброблюваних сигналів.

Ключові слова: САПР АСУТП СІНТАР, softlogic, програмуємий контролер, програмне забезпечення, сигнал, база даних, автоматизація введення даних.

The automation methods in SINTAR environment

Data input and generation methods in SINTAR softlogic environment are described. Data input method classification is introduced. In addition to conventional methods the nontraditional methods as nested iteration for multilevel structure generation, program unit SQL templates, automatic signal generation, combination of typical unit and automatic signal generation, program localization with processed signal replacement are described.

Keywords: SINTAR industrial control CAD, softlogic, programmable controller, software, signal, database, data input automation.