

СИНТАР-3 и СИНТАР 2007: сравнительный анализ систем разработки контроллерного ПО

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

Анализ новых требований

СИНТАР 2007 является новым членом семейства СИНТАР и наследует основные идеи, технические решения и программные компоненты двух непосредственных предшественников: СИНТАР-3 [1] и СИНТАР-3М [2]. Поэтому здесь рассматриваются в основном новые требования по отношению к среде СИНТАР-3, поскольку именно эта САПР была внедрена и успешно использована в ряде работок ХГПЗ им. Шевченко и его смежников на протяжении последних пяти лет.

Новые требования к АСУ в целом могут быть сформулированы таким образом:

1. Существенное увеличение масштаба АСУ (до 100 шкафов).
2. Повышение надежности и живучести системы.
3. Повышение автономности и модифицируемости элементов АСУ.
4. Централизация опроса датчиков и ввода входных данных.

Для исполнения этих требований приняты следующие структурные решения:

1. Применить в структуре шкафов управления (ШУ) более мощные контроллеры для реализации основных алгоритмов управления и контроля, а также микроконтроллеры АРМ для решения задач обмена с устройствами связи с объектом и первичной обработки сигналов.
2. Для управления работой ПО контроллеров использовать ОСРВ QNX, для управления работой ПО АРМ – диспетчер собственной разработки.
3. Использовать трехкратное резервирование на уровне ШУ.
4. Выделить для ввода аналоговые данные с датчиков одного троированного ШУ и передавать эти данные ШУ-потребителям по локальной сети.

С учетом возможного переноса в новую среду разработки старых проектов на САПР эти требования проектируются таким образом:

1. Визуальное построение сетевой многоуровневой структуры АСУ со средствами определения полных сетевых и аппаратных адресов всех элементов и размещение на них программных разделов и сигналов.
2. Предоставление пользователю средств оперативного создания и изменения структуры БД сигналов, а также описания алгоритмов автогенерации сигналов.
3. Обеспечение многоплатформенности: возможность создания ПО для разных ОС (MS DOS, QNX и диспетчера АРМ) на одном из выходных языков – Free Pascal или C++.
4. Реализация задач под QNX в виде автономных приложений с возможностью замены отдельных задач без остановки работы ШУ.

Рассмотрим эти требования более детально.

Концепция разработки

Среда СИНТАР-3 никак не связывала между собой базы данных, относящиеся к одной АСУ. Предполагалось, что это придаст известную гибкость при многократном применении ранее разработанных компонентов. Однако на деле это не принесло ожидаемого преимущества, а недостаток – налицо: если система «не знает», какие БД относятся к определенному АСУ, то эта забота ложится на пользователя. В условиях одновременной разработки и эксплуатации множества АСУ большим коллективом разработчиков это может привести к серьезной путанице.

Поэтому в среде СИНТАР 2007 для программ и данных пользователя введена новая структурная единица высшего уровня – *разработка*. Это замкнутая информационная структура, которая соответствует целой АСУ или ее части, разрабатываемой отдельной организацией или подразделением. Обмен между организациями может вестись только на уровне разработок.

Разработка представляет собой папку, которая включает:

- несколько (по числу шкафов управления) баз данных сигналов;
- несколько программных проектов;
- одну базу данных конфигураций;
- файл **Development.xml** с перечислением свойств и компонентов разработки.

Главная программа предоставляет полный набор операций над разработками, а также обеспечивает унифицированный пользовательский интерфейс для всех компонентов САПР. Редактор сигналов по указанным причинам также включен в этот интерфейс.

База данных сигналов с управляемой структурой

Эксплуатация СИНТАР-3 показала, что ориентация на фиксированную структуру БД сигналов является одним из наиболее узких мест технологии. Это не удивительно: поскольку система управления процессами фактически является системой обработки сигналов, то база данных сигналов – это объект наиболее частых изменений. Проблема усугубляется необходимостью сопровождения ранее разработанных АСУ, находящихся на разных стадиях «эволюции».

Главное отличие нового редактора сигналов: пользователь самостоятельно определяет структуру базы данных (типы сигналов и их свойства) и при возникновении необходимости может ее изменять без потери данных. Программы редактора сигналов воспринимают БД любой структуры без доработок.

Автоматическая генерация сигналов также полностью определяется пользователем. Для сигналов любого типа он может перечислить типы автогенерируемых подчиненных сигналов и указать, каким образом и при каких условиях сервер БД должен реагировать на вставку, изменение или удаление основного сигнала.

Как и прежде, каждый шкаф управления имеет свою БД сигналов. Однако увеличение объема передаваемых сигналов между шкафами потребовало включения возможности экспорта выбранных сигналов из одной БД в другие. Свойства импортированных (внешних) сигналов можно изменять только в БД-источнике импорта. При открытии некоторой БД свойства внешних сигналов обновляются автоматически, синхронизируясь с БД-источника.

Принцип автоматической синхронизации поддерживается также редактором программ и генератором. Свойства используемых в программном проекте сигналов при открытии проекта редактором программ или генератором читаются непосредственно из БД сигналов, в которой они были созданы: внешние - из БД-источника импорта, внутренние – из текущей БД. Это избавляет от используемой в СИНТАР-3 команды сопоставления сигналов в программном проекте и БД сигналов.

Для использования в алгоритмах не только текущего значения сигнала, но и его свойств, описанных в БД, пользователь может описать сигнал как некоторую структуру и обращаться к соответствующим полям. Это исключает необходимость изменения программ при изменении свойств в БД сигналов: значения свойств наряду со значением сигнала заполняются из сервера данных контроллера с помощью программ системного ПО.

Для фильтрации с целью удобства просмотра и редактирования сигналов предусмотрена возможность их классификации. Пользователь вводит произвольное количество ключей, определяющих одну или несколько классификаций, например, по регуляторам, или «входной», «выходной», «программно-формируемый», «уставка», связывает ключи с сигналами и затем может задать фильтр как комбинацию некоторых ключей с использованием логических связей AND, OR, NOT. Свойства сигналов и их значения также могут быть включены в выражение фильтрации.

Визуальное построение конфигураций

Конфигуратор СИНТАР 2007 объединяет функции прежних конфигуратора и редактора аппаратуры. Он способен обслуживать произвольное количество программно-аппаратных конфигураций, каждая из которых включает полную информацию о структуре и топологии локальной сети АСУ, размещении программных компонентов на контроллерах, распределении сигналов по линиям связи.

Конфигурация представляется в двух формах:

- древовидная структура аппаратных средств;
- иерархия вложенных схем.

Дерево аппаратных средств охватывает вершины всех уровней, начиная от уровней конфигурации, подсистемы, ШУ и заканчивая «посадочным местом» для одного сигнала. Вершина дерева в зависимости от типа может быть элементом нескольких сетей и, соответственно, имеет несколько сетевых адресов либо некоторый аппаратный адрес, идентифицирующий ее положение в пространстве аппаратных адресов данной конфигурации.

Интерфейс построения дерева позволяет:

- описывать свойства создаваемых вершин «вручную»;
- определять автоматическую генерацию вложенных вершин нескольких уровней с помощью шаблонов;
- создавать и многократно использовать типовые вершины-устройства.

Пользователь строит дерево конфигурации и определяет связи между вершинами. По этой информации конфигуратор автоматически рисует схемы всех уровней, на которых в виде вершин отражены схемы более низкого уровня, программируемые узлы сети и устройства, а также связи между ними. С помощью удобного интерфейса пользователь «заселяет» связи сигналами, выбираемыми из БД, возможно, с применением фильтрации.

В отличие от СИНТАР-3 редактор задач теперь включен в конфигуратор. Этим подчеркивается, что задача является конфигурационным, а не программным понятием [3]. Поскольку каждая ОС приписывает задачам свой набор свойств, интерфейс редактора задач изменяется в зависимости от выбранной пользователем платформы (MS DOS, QNX или ОС APM).

При размещении программ на узлах сети пользователь выбирает задачи, связывает их с узлами и наполняет разделами, выбирая последние из различных программных проектов. Минимальным размещаемым элементом ПО является не задача, как в СИНТАР-3, а программный раздел, объем которого пользователь определяет по своему усмотрению. Это дает возможность собирать программу одной задачи из проектов, разработанных на разных рабочих местах и, возможно, в разных организациях.

Таким образом, конфигуратор позволяет связать между собой аппаратуру, программное обеспечение и сигналы, а также получить визуальное представление программно-аппаратной структуры и локальной сети разрабатываемой АСУ.

Многоплатформенная генерация программ

Новый генератор обеспечивает генерацию исходных текстов на одном из двух выходных языков, объектом Паскале или С++ для работы на одной из трех платформ: MS DOS, QNX или ОС APM.

Генератор разделен на две части:

- загрузку из баз данных, не зависящую от платформы;
- генератор кодов, зависящий от платформы и выходного языка.

Используя информацию из БД конфигураций, генератор создает исходные тексты для одного выбранного узла (платформа MS DOS/APM) или одной задачи выбранного узла (платформа QNX), извлекая заданные разделы из указанных пользователем программных проектов.

Для удобства ручной настройки со стороны пользователя при генерации некоторых файлов (например, главный файл задачи или make-файл) используются шаблоны. Это позволяет изменять содержание выходного файла без модификации генератора.

Скорость генерации по сравнению с генератором СИНТАР-3 возросла в 2-3 раза.

Новая технология выполнения работ

Технология разработки любого продукта, не только программного, представляется как частично упорядоченное множество отдельных работ. Частичная упорядоченность означает, что для произвольно взятой пары работ либо существует ограничение на порядок выполнения, либо нет. Во втором случае работы этой пары могут быть выполнены в произвольном порядке, в том числе параллельно. Эта частичность порядка создает известную неразбериху, но в то же время дает некоторую свободу маневра.

Разработка реального продукта включает не только первоначальное создание, но и последующие доработки или модификации. Доработка, как правило, захватывает небольшую часть работ из общего множества работ данной технологии и тем самым нарушает общую последовательность. Тем не менее эта общая последовательность (возможно, с параллельными ветвями) должна существовать,

по крайней мере, для первоначального создания продукта. Можем назвать ее *основным потоком работ*.

Если перейти к технологии разработки ПО, поддерживаемой некоторой САПР, то работа – это некая команда или группа связанных команд пользовательского интерфейса, а основной поток работ – это тот стержень, на котором строится логика САПР.

Какие вообще работы выполняет САПР? Это работы двух видов: либо Вы вводите данные, которые она размещает в своих БД (это делают *редакторы*), либо Вы получаете на основе введенных данных некоторые преобразованные данные (это делают *генераторы*).

Хорошая технология не должна допускать дублирования ввода данных, по крайней мере, в основном потоке работ. При разработке большой системы дублированные работы выполняют разные люди, что не только увеличивает затраты труда, но и приводит к многочисленным несоответствиям.

Если применить в СИНТАР 2007 (укрупненный) основной поток работ, поддерживаемый СИНТАР-3, то с учетом структурных нововведений он бы выглядел так:

1. Создать разработку.
2. Создать в ней базы данных сигналов (по одной на ШУ).
3. Заполнить каждую БД сигналами.
4. Экспортировать межстоечные сигналы в другие БД разработки.
5. Создать программный проект.
6. Импортировать в него сигналы из некоторой БД.
7. Разработать исходники ПО с помощью редакторов САПР.
8. Повторить пункты 5 - 7 необходимое количество раз.
9. Создать конфигурацию.
10. Построить дерево аппаратуры (включающее ШУ).
11. Ввести связи между вершинами дерева (в том числе межстоечные).
12. Связать вершины дерева с программами.
13. Наполнить сигналами связи между вершинами.
14. Получить исполняемые файлы для (всех задач (QNX)) каждого узла.
15. Отладить конфигурацию.

Поскольку мы разбили общую БД сигналов на части, равные шкафам, то получается, что одна БД сигналов однозначно соответствует одному ШУ, так сказать, по определению. Поэтому работа 2 вносит в систему информацию о шкафах. Эта информация частично дублируется работой 10.

Работа 4 подразумевает знание межстоечных связей, и потому работа 11 частично дублирует информацию, полученную системой при выполнении работы 4.

Работа 6 устанавливает связь между БД сигналов (т.е. шкафом) и программным проектом. Та же информация повторно вводится работой 12.

Рассмотрим вариант основного потока работ, принятый в среде СИНТАР 2007:

1. Создать разработку.
2. Создать конфигурацию.
3. Построить дерево аппаратной структуры конфигурации (включающее ШУ). Создать базу данных сигналов ШУ (диалог появляется автоматически при создании ШУ и нужен только для указания БД-источника в случае копирования структуры БД).

4. Ввести связи между вершинами дерева (в том числе межстоечные).
5. Заполнить каждую БД сигналами.
6. Экспортировать межстоечные сигналы в другие БД разработки (другие БД определяются наличием связей между ШУ, определенных в 4).
7. Создать программный проект.
8. Импортировать в него сигналы из некоторой БД.
9. Разработать исходники ПО с помощью редакторов САПР.
10. Повторить пункты 7 - 9 необходимое количество раз.
11. Связать вершины дерева с программами (из проектов, связанных с данным ШУ через БД сигналов, см. п. 8).
12. Наполнить сигналами связи между вершинами (учитывая связи БД с ШУ, см. п. 3).
13. Получить исполняемые файлы для (всех задач (QNX)) каждого узла.
14. Отладить конфигурацию.

В этом потоке данные не дублируются, а только уточняются с учетом ранее выполненных работ. Это сокращает общую трудоемкость и снижает вероятность ошибок.

Выводы

СИНТАР 2007 представляет собой SoftLogic-систему, рассчитанную на разработку крупномасштабных АСУ. Примерами областей применения могут служить АСУ атомных электростанций, магистральных продуктопроводов, транспортных систем.

СИНТАР 2007 поддерживает объектно-ориентированную технологию программирования, использование типовых программно-аппаратных структур, многоплатформенность и многозадачность. Все основные этапы разработки основаны на визуальном представлении проектируемых объектов.

Среда оснащена рядом специфических средств автоматизации, обусловленных потребностями крупной АСУ. Она поддерживает характерный для таких АСУ коллективный характер разработки с участием нескольких организаций.

Список литературы

1. Сухоробрый В.Г., Гристан А.С., Джулгаков В.Г., Джулгаков Д.В.. САПР СИНТАР-3. Начало реализации концепции построения перспективных САПР систем сбора информации и управления технологическими процессами //Открытые информационные и компьютерные интегрированные технологии. – Х.: НАКУ «ХАИ». - 2004. – Вып. 21. – С. 83-91.
2. Сухоробрый В.Г., Гристан А.С., Джулгаков В.Г., Руденко К.И., Джулгаков Д.В., Иванов В.Е.. СИНТАР-3 Моно – САПР для одномашинной АСУ //Открытые информационные и компьютерные интегрированные технологии. – Х.: НАКУ «ХАИ». – 2004. - Вып. 24. – С. 257 - 262.
3. International standard 1131-3, Part 3: Programming languages, IEC, Division Automatismes Programmables, First edition, 1993.