

ШАБЛОН ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ УСТРОЙСТВ СВЯЗИ С ОБЪЕКТОМ АВИАЦИОННЫХ ТРЕНАЖЕРОВ

В.А. Хоменко, Е.Н. Сидоров, И.Б. Мендзевровский

Национальный авиационный университет,
03058, проспект Комарова, 1.
тел: 406 7396, факс (044) 497 8106,
E-mail: khomenkova@csfnau.kiev.ua

Предлагается шаблон программного обеспечения для построения устройств связи с объектом наследуемых авиационных тренажеров. Приводится практический пример применения шаблона при реинженерии авиационного тренажера.

Software pattern for the creation of the legacy aircraft simulators object communication device is proposed. Case study of the pattern usage for the aircraft simulator reengineering is presented.

Введение

Авиационные тренажеры – сложные технические комплексы, которые содержат множество аппаратных и программных компонентов, подверженных в процессе эксплуатации физическому износу и моральному старению [1].

Одним из компонентов тренажера, наиболее уязвимым с точки зрения износа и старения и требующим замены вместе с заменой вычислительного комплекса, является устройство связи с объектом (УСО), выполняющее задачу преобразования сигналов от органов управления кабины тренажера и пульта инструктора во входные цифровые параметры для вычислителя и обратную задачу преобразования выходных параметров вычислителя в управляющие сигналы для приборов кабины и исполнительных устройств тренажера (рис. 1). Основными причинами, вызывающими необходимость замены УСО, являются прекращение производства его компонентов и запасных частей и несовместимость с интерфейсами вычислителя при его замене.

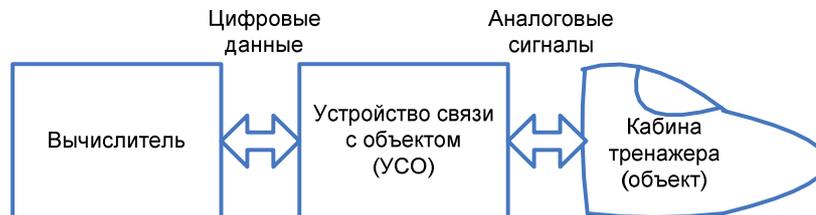


Рис. 1. Обмен данными между компонентами тренажера

При восстановлении работоспособности авиационных тренажеров создание нового УСО и соответствующего ПО является типичной задачей. Так как УСО разных авиационных тренажеров решают аналогичные задачи, то целесообразно использовать для создания их программного обеспечения шаблон, адаптируемый под конкретный тренажер.

В работе рассматривается подход к построению программного обеспечения УСО при восстановлении работоспособности авиационных тренажеров, сущность которого состоит в разработке программного шаблона. Применение шаблона демонстрируется на практическом примере создания УСО для авиационного тренажера самолета L410.

Архитектура УСО

В 70-90 годы типичный УСО авиационного тренажера строился как отдельный аппаратный или аппаратно-программный комплекс, включающий аналого-цифровые преобразователи (АЦП), цифроаналоговые преобразователи (ЦАП), преобразователи логических сигналов (ПЛС) и преобразователи для сигналов специального вида, например фазовых. Поскольку количество входных и выходных сигналов УСО обычно составляет значительное число – сто и более, то УСО отличались большой сложностью, громоздкостью и дороговизной. Современные технологии позволяют строить УСО на базе относительно недорогих промышленных компьютеров, укомплектованных необходимым набором плат устройств ввода-вывода (УВВ), устанавливаемых в системные разъемы общей шины компьютера. Управление вводом-выводом осуществляется специальным ПО (рис. 2). Большинство современных УВВ имеют собственные управляющие контроллеры, программируемые через системную шину компьютера. Кроме собственно функций ввода-вывода такие УВВ

предоставляют возможности конфигурирования, настройки режимов работы, диагностики и калибровки. Как правило, платы УВВ поставляются производителем вместе с комплектом низкоуровневых драйверов, реализующих базовые функции управления.

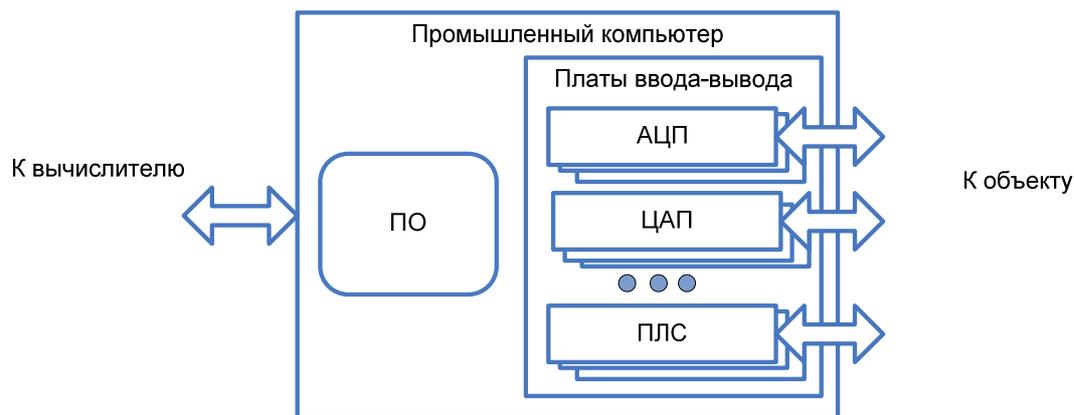
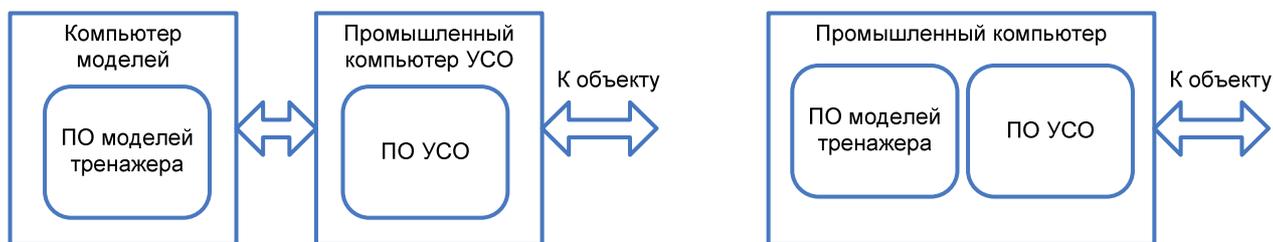


Рис. 2. Состав современного УСО

В вычислительной системе тренажера для УСО может быть выделен отдельный вычислитель (рис. 3, а) или использоваться один вычислитель для УСО и моделей тренажера (рис. 3, б).



а – на отдельном компьютере

б – совместно с моделями тренажера

Рис. 3. Варианты функционирования УСО

Наследуемые УСО, как правило, представляют собой устройства с чисто аппаратной реализацией, управляемые внешним вычислителем или имеющие управляющее ПО, написанное на ассемблере. Сейчас аппаратное обеспечение позволяет использовать в УСО операционные системы (ОС) и ПО, созданное с помощью языков высокого уровня (рис. 4).

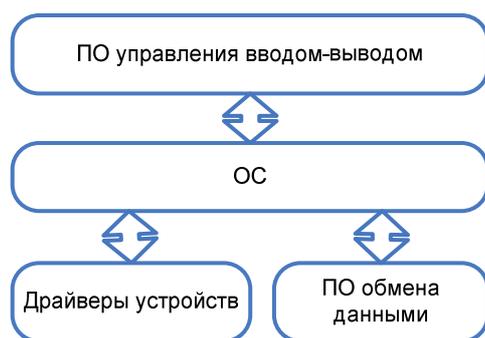


Рис. 4. Общая архитектура ПО УСО

ОС позволяет повысить эффективность разработки ПО УСО путем использования стандартных функций интерфейса прикладного программирования API и программ драйверов, поставляемых производителями УВВ. Поскольку УСО, являясь системой реального времени, должно обеспечивать гарантированное время обмена с объектом, то в составе ПО можно использовать только ОС реального времени или с минимальными затратами времени на системные процессы. Так как при функционировании УСО на вычислителе выполняется только один процесс, то от ОС не требуется обеспечения многозадачности, диспетчеризации процессов и подкачки. Эти условия делают целесообразным использование в УСО ОС типа Microsoft DOS.

ПО управления вводом-выводом предназначено для управления УВВ УСО при осуществлении обмена между объектом и моделями тренажера.

Драйверы устройств обеспечивают базовые функции управления УВВ.

ПО обмена данными обеспечивает обмен данными УСО с другими вычислительными узлами в распределенной вычислительной системе тренажера. В случае если все ПО функционирует на одном вычислителе, то обмен данными между компонентами ПО производится через память.

Шаблон для построения ПО УСО

Любое УСО должно выполнять следующие функции: обмен аналоговыми сигналами с объектом; обмен параметрами с моделями тренажера; обмен данными с другими вычислительными узлами тренажера. При этом должны обеспечиваться следующие свойства УСО:

- быстрая установка и наладка;
- коммутация сигналов на произвольные каналы УВВ и их настройка;
- эффективный интерфейс между моделями тренажера и УСО;
- защита приборов кабины и исполнительных устройств тренажера от перегрузок при подаче ошибочных значений выходных параметров.

Шаблон, исходя из выполняемых УСО функций и обеспечиваемых свойств, должен удовлетворять следующим требованиям:

- гибкая настройка и конфигурирование УВВ;
- гибкая коммутация и настройка каналов ввода-вывода;
- простое и единообразное сопряжение параметров моделей тренажера с УСО;
- поддержка интерфейсов обмена или компьютерной сети;
- ограничение предельных выходных сигналов.

Для построения УСО, удовлетворяющего этим требованиям, предлагается архитектура программного шаблона, показанная на рис. 5. Она включает два уровня драйверов ввода-вывода, которые обеспечивают функции передачи/приема параметров объекта, конфигуратор, обеспечивающий конфигурирование УВВ, входных и выходных параметров, и ПО обмена данными, реализующее взаимодействие с другими вычислительными узлами тренажера.



Рис. 5. Архитектура шаблона ПО УСО

Кроме того, для обеспечения процессов реинженерии, в архитектуре предусмотрен набор соответствующих программных инструментов, позволяющих получать информацию об объекте в процессе обратной инженерии и обеспечивать отладку и тестирование ввода-вывода параметров в процессе прямой инженерии.

Драйверы

Все параметры по признаку направления передачи делятся на входные, передаваемые от объекта к моделям и выходные, передаваемые от моделей к объекту; по признаку дискретности параметры делятся на аналоговые, непрерывно изменяющиеся во времени (положение ручек управления двигателем, положение штурвала, высота полета и т.п.) и логические (так называемые разовые сигналы), имеющие два состояния – включено и выключено (старт двигателя, состояния тумблеров автоматов защиты питания и т.п.). Для каждого типа параметров необходим свой драйвер.

Задача обмена аналоговыми сигналами с объектом и параметрами с моделями тренажера распределена между драйверами двух уровней – нижнего и верхнего. Драйверы нижнего уровня (драйверы УВВ) обеспечивают базовые функции для конфигурирования, настройки, диагностики и использования УВВ (рис. 6).

Обычно, такие драйверы поставляются разработчиком УВВ и обеспечивают обмен цифровыми значениями через вызовы соответствующих функций (инициализация, чтение, запись) с указанием номера канала платы.

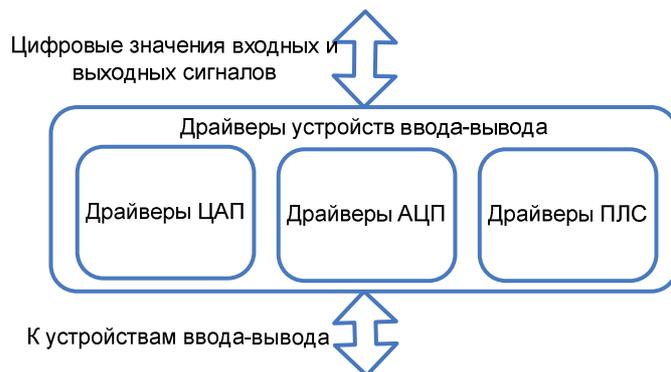


Рис. 6. Драйверы нижнего уровня

Драйверы верхнего уровня (драйверы параметров объекта) обеспечивают сопряжения УСО с параметрами моделей тренажера путем нормирования значений входных и выходных параметров (рис. 7).

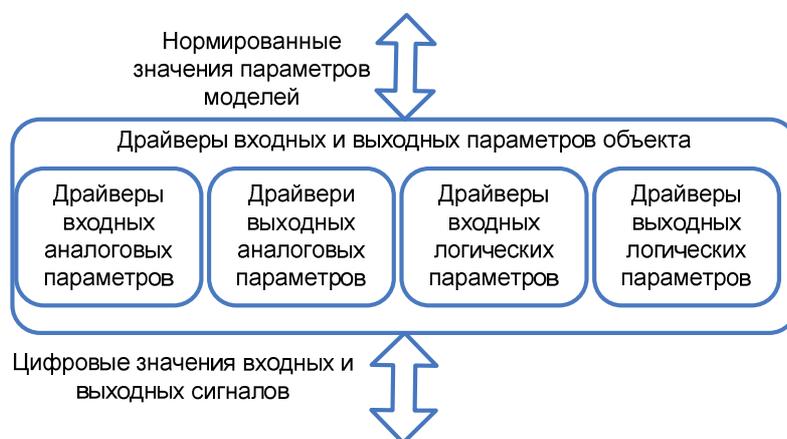
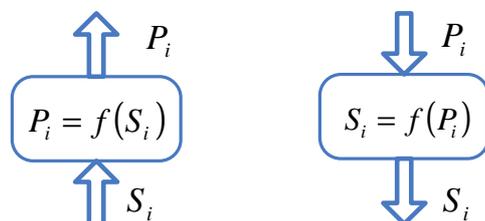


Рис. 7. Драйверы верхнего уровня

Драйверы входных параметров обеспечивают реализацию функций преобразования оцифрованных значений сигналов, принимаемых от драйверов УВВ, в значение соответствующего ему параметра модели $P_i = f(S_i)$ (рис. 8, а). Драйверы выходных параметров обеспечивают реализацию функций преобразования параметров модели в оцифрованные значения сигналов, передаваемые затем драйверам УВВ $S_i = f(P_i)$ (рис. 8, б).



а – для входных параметров б – для выходных параметров

Рис. 8. Преобразование значений драйверами параметров

Функция преобразования значений входного параметра обеспечивает линейное преобразование цифровых значений параметра, поступающих с устройства ввода, в нормированные значения параметра модели тренажера (рис. 9).

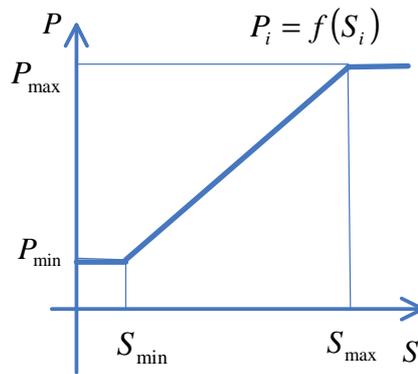


Рис. 9. Общій вид графика функции преобразования входного параметра

Функция преобразования входного параметра имеет следующий вид:

$$P_i = \begin{cases} P_{\min}, & \text{при } S_i < S_{\min}; \\ \frac{P_{\max} - P_{\min}}{S_{\max} - S_{\min}} \cdot S_i + P_{\min} & \text{при } S_{\min} \leq S_i \leq S_{\max}; \\ P_{\max}, & \text{при } S_i > S_{\max}, \end{cases} \quad (1)$$

где P_i – значение нормированного параметра модели; S_i – значение сигнала; P_{\min}, P_{\max} – минимальные и максимальные значения нормированного параметра модели; S_{\min}, S_{\max} – минимальные и максимальные значения сигнала.

Функция преобразования значения выходного параметра состоит в линейном преобразовании нормированных значений параметра, поступающих из моделей тренажера, в цифровые значения, подаваемые на устройство вывода (рис. 10).

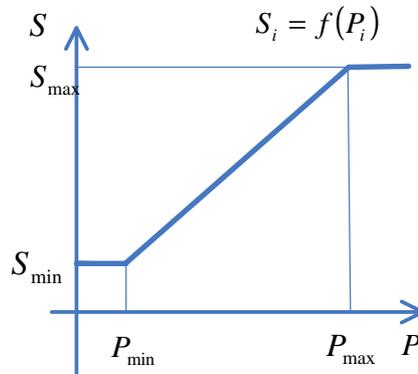
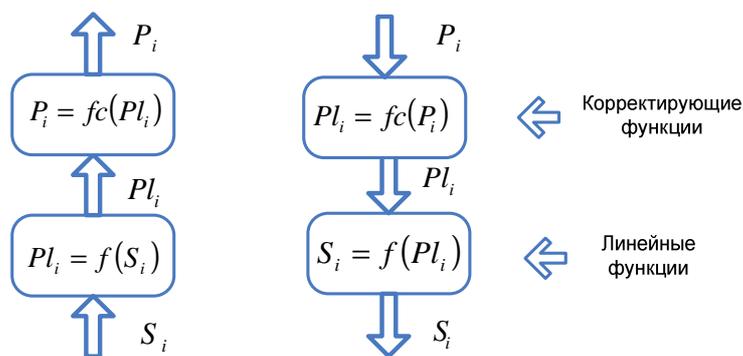


Рис. 10. Общій вид графика функции преобразования выходного параметра

Функция преобразования выходного параметра аналогична (1) и имеет следующий общий вид:

$$S_i = \begin{cases} S_{\min}, & \text{при } P_i < P_{\min}; \\ \frac{S_{\max} - S_{\min}}{P_{\max} - P_{\min}} \cdot P_i + S_{\min} & \text{при } P_{\min} \leq P_i \leq P_{\max}; \\ S_{\max}, & \text{при } P_i > P_{\max}. \end{cases} \quad (2)$$

В большинстве случаев для преобразования параметров используется линейная зависимость между входом и выходом функции, но иногда возникает необходимость реализовать нелинейное преобразование значений, например, для компенсации нелинейности датчиков или управления логарифмическими приборами. В таких случаях предлагается нормированные значения параметров дополнительно преобразовывать корректирующими функциями f_c специального вида (рис. 11).



а – для входных параметров б – для выходных параметров

Рис. 11. Преобразование значений драйверами параметров с использованием корректирующих функций

Тогда, драйверы параметров реализуют для преобразования значений следующие функционалы:

- для входных параметров – $P_i = fc(f(S_i, S_{\min}, S_{\max}, P_{\min}, P_{\max}))$;
- для выходных параметров – $S_i = fc(f(P_i, S_{\min}, S_{\max}, P_{\min}, P_{\max}))$.

– В предлагаемом шаблоне для конфигурирования и настройки каждого из входных и выходных параметров необходимо произвести следующее:

- задать тип параметра (входной или выходной, непрерывный или логический);
- связать параметр с каналом ввода (вывода), путем задания идентификатора УВВ и номера одного из его каналов;
- определить линейную функцию преобразования значений параметра путем задания пределов изменения нормированного значения параметра и цифрового значения, подаваемого или получаемого с УВВ;
- определить корректирующую функцию параметра.

Каждый сконфигурированный параметр УСО получает идентификатор, который затем используется как аргумент в соответствующих функциях ввода и вывода.

Конфигуратор

Для обеспечения эффективного конфигурирования УСО шаблон ПО включает конфигураторы УВВ и параметров, а также хранилище данных конфигурации (рис. 12).

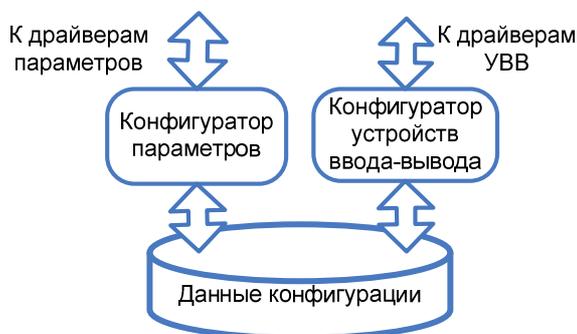


Рис. 12. Конфигуратор

Конфигуратор УВВ осуществляет настройку устройств в соответствии со значениями, содержащимися в хранилище данных конфигурации. К таким настройкам относятся адреса, номера прерываний и каналов прямого доступа к памяти, используемые УВВ. Конфигуратор при запуске ПО УСО осуществляет считывание значений настроек из хранилища данных конфигурации, конфигурирование, инициализацию и диагностику устройств, используя функции их драйверов, и создает системные структуры данных, содержащие параметры конфигурации УВВ.

Конфигуратор параметров осуществляет назначение каналов ввода или вывода параметрам моделей тренажера и выполняет соответствующие настройки. В частности, для каждого параметра задается идентификатор УВВ, номер канала, граничные значения сигналов и нормированных параметров, а также, при необходимости, корректирующая функция. При инициализации УСО, значения настроек, содержащиеся в хранилище данных конфигурации, считываются конфигуратором для использования драйверами параметров при осуществлении операций ввода-вывода. Граничные значения сигналов УСО удобнее определять не в

цифровых значениях, а в соответствующих значениях напряжения на аналоговых выводах устройств, поэтому конфигуратор должен обеспечивать пересчет хранимых граничных значений напряжений в цифровые значения.

Для организации хранилища данных конфигурации целесообразно применять язык XML. Это позволяет создавать иерархическое дерево параметров, которое можно редактировать, просматривать и использовать стандартными средствами работы с XML. Для применения XML в конфигураторе определен специальный язык, основные метки которого даны в таблице.

Таблица

Метка	Назначение
Conf	Начало конфигурации
ConfName	Название конфигурации
Cards	Раздел конфигурации устройств ввода-вывода
Card	Конфигурация УВВ
CardType	Тип устройства
CardNum	Номер устройства
ChanQual	Число каналов устройства
CardAddress	Адрес устройства
CardIRQ	Номер прерывания
CardMode	Режим работы
Parameters	Раздел конфигурации параметров
IParam	Раздел входных параметров
OParam	Раздел выходных параметров
Analog	Раздел аналоговых параметров
Logical	Раздел логических параметров
Param	Конфигурация параметра
ParamName	Название параметра
ParamNum	Номер параметра
Pmin	Минимальное нормированное значение параметра
Pmax	Максимальное нормированное значение параметра
Smin	Минимальное значение сигнала
Smax	Максимальное значение сигнала
ChanNum	Номер канала

Пример описания конфигурации одного из УВВ и входного параметра показан на рис. 13.

```

<Configuration>
  <ConfName> L410 USO </ConfName>
  <Cards>
    <Card>
      <CardType> ADO 32 </CardType>
      <CardNum> 0 </CardNum>
      <ChanQual> 32 </ChanQual>
      <CardAddress> 220 </CardAddress>
      <CardIRQ> none </CardIRQ>
      <CardMode> default </CardMode>
    </Card>
  </Cards>
  <Parameters>
    <IParam>
      <Analog>
        <Param>
          <ParamNum> 0 </ParamNum>
          <ParamName> RUD </ParamName>
          <Pmin> 0 </Pmin>
          <Pmax> 1 </Pmax>
          <Smin> 0 </Smin>
          <Smax> 0.93 </Smax>
          <CardNum> 0 </CardNum>
          <ChanNum> 0 </ChanNum>
        </Param>
      </Analog>
    </IParam>
  </Parameters>
</Configuration>

```

Рис. 13. Пример описания конфигурации устройства ввода-вывода и параметра

Для чтения и загрузки данных конфигурации из XML-файла в конфигураторе используется XML-анализатор [2].

ПО обмена данными

ПО обмена данными обеспечивает прием и передачу значений параметров моделей тренажера через компьютерную сеть или интерфейс обмена. Обмен между узлами на прикладном уровне должен реализовываться путем формирования в приложениях-отправителях сообщений, содержащих набор значений параметров тренажера, передачей их по сети, приемом и доставкой в приложения-получатели.

Сообщения могут иметь жесткую структуру, включающую значения параметров, посылаемые строго между двумя получателями на определенный адрес и порт и гибкую, включающую, кроме значений параметров их идентификаторы, адреса отправителя и получателя сообщения. Первый тип сообщений обеспечивает простоту реализации и экономное расходование трафика, а второй – гибкость и самодокументируемость реализации. Для реализации сообщений второго типа предлагается использовать XML-строки, имеющие формат, показанный на рис. 14.

<HD>	<T>	ID получателя	</T>	<S>	ID отправителя	</S>	</HD>	<P>	<PN>	ID параметра	</PN>	<V>	Значение	</V>	...	</P>
------	-----	---------------	------	-----	----------------	------	-------	-----	------	--------------	-------	-----	----------	------	-----	------

Рис. 14. Формат сообщений

ПО обмена данными должно содержать средства для формирования исходящих сообщений и разбора входящих, при этом целесообразно использовать единую сквозную идентификацию параметров, что позволит использовать хранилище конфигурации, как для драйверов параметров, так и для средств обмена данными.

Процесс и инструменты реинженерии УСО

Процессы построения УСО с помощью шаблона соответствуют процессам реинженерии наследуемого ПО и делятся на две фазы (рис. 15) [3, 4]. Первая – соответствует обратной инженерии и направлена на получение информации о наследуемом тренажере, необходимой для построения УСО с помощью шаблона. Вторая – соответствует прямой инженерии и направлена на конфигурирование, настройку и отладку УСО. В результате выполнения первой фазы выделяется следующая информация:

- набор входных и выходных параметров и их характеристики;
- характеристики датчиков, приборов и исполнительных устройств тренажера.

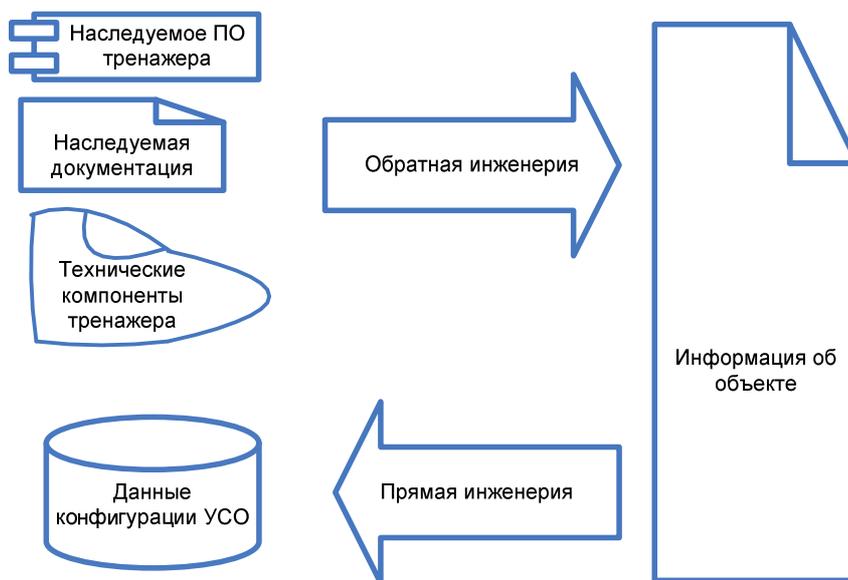


Рис. 15. Фазы реинженерии УСО

Информация может быть получена следующими способами: анализом существующей документации; созданием высокоуровневых представлений наследуемого ПО; экспериментальным исследованием поведения компонентов тренажера.

Анализ существующей документации является наиболее эффективным способом получения информации о тренажере, однако обычно невозможен вследствие отсутствия документации.

Создание высокоуровневых представлений ПО производится путем анализа исходных и машинных текстов и дает наиболее точную информацию о логике работы наследуемого ПО, но не позволяет получить текущие характеристики аналоговых компонентов тренажера.

Экспериментальные исследования поведения компонентов тренажера производятся путем подачи на них тестовых воздействий с последующей фиксацией и исследованием их откликов. Для реализации такого исследования предлагается использовать в составе шаблона специальный набор инструментов, состоящий из измерителей и генераторов сигналов.

Измерители сигналов предназначены для снятия характеристик органов управления и датчиков тренажера (рис. 16). Они обеспечивают прием цифровых значений сигналов или нормированных параметров, расчет напряжений, соответствующих полученным значениям, визуализацию значений в виде чисел и графиков.

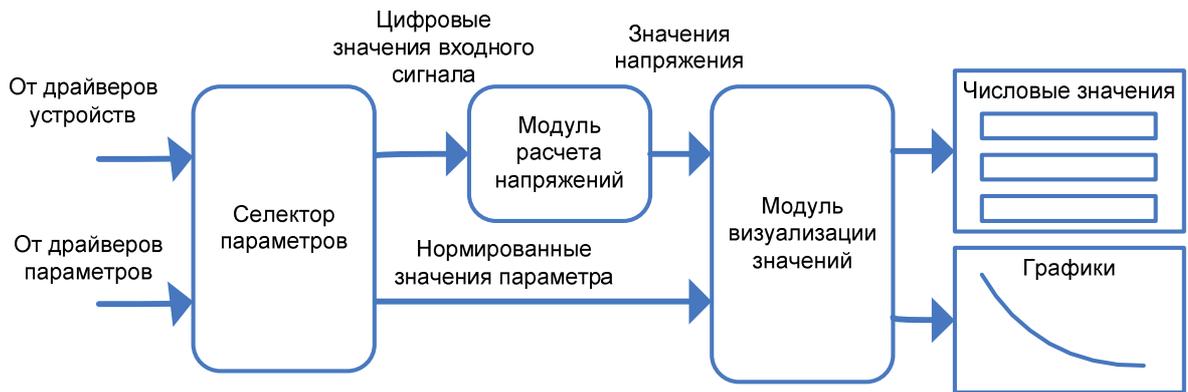


Рис. 16. Структура ПО измерителя сигналов

Генераторы сигналов предназначены для подачи тестовых сигналов на индикаторные приборы и исполнительные устройства тренажера (рис. 17). Оператор имеет возможность задать для управления выходом устройства любое из трех типов значений – напряжение сигнала, соответствующее ему цифровое значение УВВ или значение нормированного параметра.



Рис. 17. Структура ПО генератора тестовых сигналов

Практический пример

Разработанный шаблон для построения УСО был реализован и проверен в Национальном авиационном университете при реинженерии авиационного тренажера TL 410 чешского производства 80-х годов [5]. Тренажер не эксплуатировался с начала 90-х годов в связи с выходом из строя вычислителя Robotron и УСО DACIO. Целью проекта было полное восстановление функционирования тренажера, за исключением подвижности кабины. Восстановление осуществлялось путем замены вычислителя, УСО и переработки программного обеспечения.

Новая архитектура тренажера включала вычислительную систему из трех вычислителей, объединенных в локальную компьютерную сеть, содержащую три узла: моделей самолета и УСО (см. рис. 3, б); визуализации и имитации шума; рабочего места инструктора.

Процесс построения УСО включал установку и конфигурирование вычислителя и УВВ (АЦП, ЦАП и ПЛС), установку и конфигурирование операционной системы MS DOS, конфигурирование УВВ, разработку драйверов параметров, определение набора и характеристик входных и выходных параметров тренажера,

снятие характеристик приборов и органов управления кабины, конфигурирование параметров и организацию обмена данными УСО с другими вычислительными узлами. В результате был полностью восстановит обмен сигналами с объектом (около 120 входных и выходных сигналов) и соответствующими параметрами с моделями тренажера. Для обмена параметрами с другими узлами были использованы стек протока TCP/IP и реализовано ПО обмена данными.

Оценка затрат на построение и конфигурирование УСО на основе шаблона показывает следующее их распределение по видам работ:

- установка и конфигурирование аппаратного обеспечения – 2 чел. дня;
- установка и конфигурирование ОС и стека сетевого протокола – 2 чел.дня;
- разработка драйверов параметров – 1 чел. мес;
- снятие характеристик приборов и органов управления кабины – 20 чел. дней;
- конфигурирование параметров – 2 чел. дня;
- организация обмена данными – 5 чел. дней.

Процесс создания и конфигурирования УСО с помощью шаблона показал высокую эффективность разработанных средств. Например, среднее время, затрачиваемое на конфигурирование и тестирование в УСО каждого из 120 параметров, составило около 10 минут. Применение многоуровневой архитектуры шаблона ПО УСО, стандартизованных интерфейсов и нормирования параметров моделей позволило полностью разделить процессы создания УСО и остального ПО тренажера.

1. *Design of a flight simulator software architecture.* Göran Ancker, Jan Wallenberg. – School of Mathematics and Systems Engineering, Växjö University. – 2002. – 91 p.
2. *Бин Д.* XML для проектировщиков. Повторное использование и интеграция. – М.: КУДИЦ-ОБРАЗ, 2004. – 256 с.
3. *Сидоров Н.А.* Восстановление, повторное использование и переработка программного обеспечения // УСИМ. – 1998. – № 3 – С. 79 – 83.
4. *Соммервил И.* Инженерия программного обеспечения. – М.: Издательский дом «Вильямс», 2002. – 624 с.
5. *Сидоров М.О., Иванова Л.М., Хоменко В.А.* Методологічні принципи реінженерії програмного забезпечення успадкованих авіаційних тренажерів // Мат. VIII Міжнар. наук.-техн. конф. „Авіа-2007”. – К.: 25-27 квітня 2007. – Т. 1, С. 13.119–13.122.