Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Маслов Никита Сергеевич

Разработка инструмента анализа и автоматической проверки требований для информационного обмена в бортовых сетях передачи данных

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
А.В.Герасёв

Москва, 2017

Содержание

Аннотация

В настоящее время в авиационных и корабельных бортовых вычислительных комплексах широко используется мультиплексный канал информационного обмена (МКИО), при этом для корректной работы комплекса передаваемые по каналу данные должны соответствовать протоколам, согласованным и утверждённым разработчиками.

Данная работа посвящена формализации описания требований, предъявляемым к обменам и передаваемым данным, а также задаче автоматизации анализа соответствия передаваемых данных протоколам, записанным в предложенном формальном виде.

В работе приведён перечень проверяемых требований, предложена архитектура решения и создано программное средство для проверки требований на основе инструмента Opermon, разрабатываемого в ЛВК.

1 Введение

1.1 Актуальность задачи

В РВС компоненты взаимодействуют друг с другом, используя согласованные и утвержденные разработчиками системы протоколы. В этих протоколах содержится информация об абонентах (компонентах системы), адресации на каналах, наборе передаваемых сообщений, частотах этих сообщений, их форматах, ограничениях на последовательность, описание содержимого этих сообщений.

Одна из задач интеграционного тестирования РВС - проверка соблюдения этих протоколов компонентами системы. Для этого в среде тестирования описываются тесты, которые выполняют соотвествующие проверки: для принятых от тестируемого компонента сообщений проверяются различные характеристики из упомянутых выше. Некоторые детали соблюдения протоколов можно проверить только так: передать компоненту сообщение, содержащее параметр N, получить от него в ответ сообщение с зависимым параметром M и проверить, что преобразование параметра компонентом выполнено правильно. С другой стороны, некоторые ограничения (например, частотные характеристики) возможно проверить только при регистрации обменов на работающей системе и сравниваи характеристик обменов с ограничениями, описанными в протоколах.

Таким образом, целесообразно иметь инструмент, который по регистрируемым обменам в реальном времени или по записанной трассе сможет проверить соответствие этих обменов требованиям протоколов.

2 Постановка задачи

2.1 Формальная постановка задачи

2.2 Требования к анализатору

К решению задачи предъявляется следующий набор требований, связанных с возможными способами прикладного применения анализатора.

- Простота: описание конфигурации анализатора должно быть простым для понимания, создания и редактирования как вручную пользователем анализатора, так и с применением программного обеспечения для автоматического составления описания протоколов.

- Совместимость: формат описания конфигурации анализатора должен быть совместимым с используемым на текущий момент форматом описания протокола для Анализатора МКИО для возможности переиспользования существующего программного обеспечения для автоматического составления описания протоколов.

- Быстродействие: анализатор должен иметь возможность выполняться достаточно быстро для работы в режиме регистрации обменов в реальном времени при условии выполнении анализа на рабочей станции пользователя Анализатора МКИО и проверки достаточно большого количества требований.

- Расширяемость: анализатор должен быть готовым к возможным расширениям функционала (определения новых типов требований к обменам и параметрам) с сохранением совместимости с предыдущей версией как минимум на уровне формата описания конфигурации. Это значит, что при добавлении новых типов ограничений пользователь должен иметь возможность использовать описания требований, использованных с предыдущей версией анализатора (возможно, с незначительными строго описанными изменениями).

В качестве дополнительного требования, для удобства работы описание конфигурации анализатора требований должно иметь возможность сохранения во внутреннем пользовательском хранилище Анализатора МКИО для последующего переиспользования без необходимости повторного импорта описания из внешнего файла.

2.3 Формат описания входных данных

Формат входных данных обратно совместим с форматом, использованным в некоторых версиях Opermon для описания сообщений и битовых полей на основе спецификации ПИВ.

Входные данные представляют собой XML-документ следующего содержания:

```
<?xml version="1.0"?>

<piv version="1.1">
   <signals>
      <!-- ...signals -->
      <signal identifier="" type="" signed="" twosComplement="" />
      <!-- ... -->
   </signals>

   <abonents>
      <!-- ...abonents -->
      <abonent identifier="" mil1553_addr="">
         <!-- ...type_messages -->
         <mil1553_messages>
            <!-- controller messages -->
            <mil1553_contrMessage identifier="" direction="" addr
               ="" subaddr="" numWords="">
               <!-- ...bitfields -->
               <bitfield identifier="" firstWord="" firstBit=""
                  numBits="" lowerBitCost="">
                  <restrict type="" value="" level="" />
                  <!-- ... -->
               </bitfield>
               <!-- ... -->
            </mil1553_contrMessage>

            <!-- terminal messages -->
            <mil1553_termMessage identifier="" direction=""
               subaddr="" numWords="">
               <!-- ...bitfields -->

            </mil1553_termMessage>
         </mil1553_messages>
         <!-- ... -->
      </abonent>
      <!-- ... -->
```

```
    </abonents>

    <restricts>
        <!-- restrics -->
        <restrict messageId="" type="" level="" value="">
            <!-- special restriction params -->
            <param name="" value="" />
            <!-- ... -->
        </restrict>
        <!-- ... -->
    </restricts>
</piv>
```

Листинг 1: Структура файла описания входных данных

Атрибуты описания абонента (тег abonent):

- identifier - индентификатор абонента (строка - идентификатор Си);
- mil1553_addr - адрес абонента на шине MIL STD-1553B.

Атрибуты описания сигнала (тег signal):

- identifier - идентификатор сигнала (строка - идентификатор Си);
- type - тип данных сигнала (например, int, unsigned int, double); приведён для справки;
- signed - является ли сигнал знаковым (true|false, по умолчанию true);
- twosComplement - записывается ли отрицательное значение в дополнительном коде (true|false, по умолчанию false для совместимости со старым форматом ПИВ).

Тег restrict также может содержать элементы внутри, если это требуется для определённого типа ограничений.

Атрибуты сообщения MIL STD-1553B для контроллера (тег mil1553_contrMessage):

- identifier - идентификатор сигнала (строка - индентификатор Си);
- direction - направление (input | output - к/от контроллера);
- addr - адрес ОУ (число от 1 до 31);
- subaddr - подадрес ОУ (целое число от 1 до 30);

- numWords - число слов в сообщении (от 1 до 32).

Атрибуты сообщения MIL STD-1553B для оконечного устройства (тег mil1553_termMessage):

- identifier - идентификатор сигнала (строка - индентификатор Си);
- direction - направление (input | output - к/от контроллера);
- subaddr - подадрес ОУ (целое число от 1 до 30);
- numWords - число слов в сообщении (от 1 до 32).

Стоит заметить, что в описании сообщений MIL STD-1553B для оконечных устройств не указан адрес ОУ-получателя сообщения. Это связано с особенностями внутреннего устройства используемых БД ПИВ. Для формирования полного заголовка сообщения требуется найти два "полусообщения" - сообщения mil1553_termMessage у двух абонентов, где атрибуты identifier для сообщений совпадают.

Атрибуты описания ограничений для сигнала (тег restrict внутри тега bitfield):

- type - тип ограничения (см. Ограничения для сигналов);
- value - значение для ограничения (необязательный параметр), зависит от типа ограничения;
- level - уровень критичности ограничения (info, notice, warning, error).

2.4 Формирование проверяющих объектов

2.5 Ограничения для сообщений

2.5.1 Частота появления сообщения

Проверяется частота появления сообщения требуемого типа на шине.

Подсчёт частоты происходит вычислением временного интервала между получением текущего и предыдущего сообщений (точное время получения сообщений записано в структуре Exchange).

Параметры ограничения:

- *value* - требуемое значение частоты в герцах (Гц);
- maxDeviation - максимальное отклонение значения частоты (по модулю). Может быть записано в абсолютной величине (без суффикса; например, 1.0), так и в процентах (с суффиксом '%'). По умолчанию - 5%.

Описание ограничения в файле протокола:

```
<restrict messageId="message_id" type="frequency" level="warning"
    value="10.0">
    <param name="maxDeviation" value="1.0" />
</restrict>
```

Листинг 2: Частота появления сообщения

2.5.2 Ошибочные состояния

Проверяются ошибочные состояния сообщения (флаги MIL STD-1553B).

Флаги описаны в структуре Exchange.

Параметров у ограничения нет.

Описание ограничения в файле протокола:

```
<restrict messageId="message_id" type="errors" level="warning" />
```

Листинг 3: Ошибочные состояния сообщения

2.5.3 Последовательность сообщений

Проверяется последовательность сообщений.

Последовательность сообщений задаётся с помощью идентификатора последовательности. Каждое сообщение, входящее в последовательность, имеет в ней порядковый номер. Анализатор проверяет, соблюдается ли порядок появления сообщений в канале согласно порядку номеров.

Стоит заметить, что сообщения не обязательно должны следовать друг за другом; между сообщениями одной последовательности могут появляться другие сообщения. Проверяется порядок именно тех сообщений, которые включены в последовательность.

В параметрах ограничения описывается строковой идентификатор последовательности (идентификатор Си) и номер сообщения в последовательности.

Если после анализа файла протокола выяснится, что номера каких-либо сообщений в последовательности совпадают, пользователь получит сообщение об ошибке и ограничение проверяться не будет. Последовательность пар номер - тип сообщения будет упорядочена по номеру.

При получении первого сообщения за время работы анализатор выставит внутренний индекс на номер полученного сообщения. При получении следующего анализатор сравнит номер следующего полученного сообщения со следующим сообщением в последовательности. При несовпадении будет выведено сообщение об ошибке, при этом внутренний индекс вновь будет сброшен.

Параметры ограничения:

- sequenceId - идентификатор последовательности (строка - идентификатор Си);
- order - номер сообщения в последовательности.

Описание ограничения в файле протокола:

```
<restrict messageId="message1" type="sequence" level="warning">
    <param name="sequenceId" value="sequence1" />
    <param name="order" value="1" />
</restrict>
<restrict messageId="message2" type="sequence" level="warning">
    <param name="sequenceId" value="sequence1" />
    <param name="order" value="2" />
</restrict>

<!-- ... -->

<restrict messageId="messageN" type="sequence" level="warning">
    <param name="sequenceId" value="sequence1" />
    <param name="order" value="N" />
</restrict>
```

Листинг 4: Последовательность сообщений

В случае несовпадения параметра level у описаний одной и той же последовательности, пользователь получит предупреждение, при этом будет выбрано самое сильное значение параметра.

2.5.4 Значение контрольной суммы в полезной нагрузке

Проверяется значение контрольной суммы для некоторого диапазона байт в полезной нагрузке сообщения.

Подразумевается, что в полезной нагрузке сообщения можно специально выделить две последовательности байт, где одна из них - блок данных, а вторая - значение контрольной суммы для этого блока данных.

Параметры ограничения:

- function - функция подсчёта контрольной суммы (crc16 на текущий момент);
- dataStart - номер первого слова последовательности блока данных;
- dataSize - длина последовательности блока данных (количество слов);
- checksumWord - номер слова поля контрольной суммы. Подразумевается, что длина поля контрольной суммы известна по функции подсчёта.

Описание ограничения в файле протокола:

```
<restrict messageId="message_id" type="checksum" level="warning">
    <param name="function" value="crc16" />
    <param name="dataStart" value="0" />
    <param name="dataSize" value="8" />
    <param name="checksumWord" value="8" />
</restrict>
```

Листинг 5: Проверка контрольной суммы

2.6 Ограничения для параметров

2.6.1 Частота обновления параметра

Проверяется минимальная частота обновления значения параметра.

Подсчёт частоты происходит вычислением временного интервала между получением текущего и предыдущего сообщений. Точное время получения значения параметра передаётся вместе со значением в структуре ParameterContainer::ParamValue.

Параметры ограничения:

- *value* - требуемое минимальное значение частоты в герцах (Гц);

Описание ограничения в файле протокола:

```
<signal identifier="signal1">
   <restrict type="minFrequency" level="error" value="1.0" />
</signal>
```

Листинг 6: Частота появления сообщения

2.6.2 Пороговые значения

Проверяется выход значения параметра за некоторое пороговое значение (вверх или вниз).

Для одного параметра можно описать несколько различных пороговых значений (в том числе одного типа) при том, что у ограничений будут различаться уровни критичности (параметры level). В случае, если значение параметра вышло за несколько пороговых значений одного типа (min или max), сообщение будет выведено для порогового значения с самым сильным значением уровня критичности.

Типы ограничений:

- min - минимальное значение параметра;
- max - максимальное значение параметра.

Параметры ограничения:

- *value* - пороговое значение.

Описание ограничения в файле протокола:

```
<signal identifier="signal1">
   <restrict type="max" level="error" value="10.0" />
   <restrict type="max" level="warning" value="9.0" />

   <restrict type="min" level="warning" value="2.0" />
   <restrict type="min" level="error" value="1.0" />
</signal>
```

Листинг 7: Пороговые значения параметра

2.6.3 Равенство константе

Проверяется равенство значения параметра определённой константе (или равенство с допустимой погрешностью).

Параметры ограничения:

- *value* - константа,
- maxDeviation - максимальное отклонение значения от константы в абсолютной величине (по модулю). По умолчанию - 0.

Описание ограничения в файле протокола:

```
<signal identifier="signal1">
    <restrict type="const" level="error" value="10.0" />
</signal>

<signal identifier="signal2">
    <restrict type="const" level="error" value="120.0">
        <param type="maxDeviation" value="2.0" />
    </restrict>
</signal>
```

Листинг 8: Равенство значения параметра константе

2.6.4 Ошибочное значение

Проверяется равенство значения параметра некоторой константе, означающей ошибочное состояние параметра. Значение должно быть целочисленным.

Параметры ограничения:

- *value* - ошибочное значение.

Описание ограничения в файле протокола:

```
<signal identifier="signal1">
    <restrict type="error_value" level="error" value="0xDEAD" />
</signal>
```

Листинг 9: Ошибочное значение параметра

2.6.5 Гладкость

Проверяется гладкость параметра - ограничение на максимальную (по модулю) скорость изменения значения параметра.

Скорость изменения параметра измеряется в единицах измерения значения параметра в секунду и считается между двумя соседними событиями обновления значения параметра по формуле:

$v = \frac{val_2 - val_1}{t_2 - t_1}$,

где v - скорость изменения значения параметра, val_2, val_1 - соответственно текущее и предыдущее значения наблюдаемого параметра, t_2, t_1 - время получения текущего и предыдущего значения параметра соответственно (в секундах с момента начала записи трассы).

Параметры ограничения:

- *value* - максимальное значение скорости изменения параметра (по модулю).

Описание ограничения в файле протокола:

```
<signal identifier="signal1">
    <restrict type="smooth" level="error" value="0.25" />
</signal>
```

Листинг 10: Гладкость значения параметра

2.6.6 Связанные параметры

Проверяется соответствие значений нескольких различных параметров, имеющих общую природу.

Например, высота над уровнем моря на борту самолёта может быть получена от модуля позиционирования, использующего GPS, и от модуля, использующего барометрический датчик. Каждый модуль предлагает свой собственный параметр, требуется сравнить эти параметры с некоторой заранее заданной погрешностью.

Параметры связываются в группы, определёнными с помощью строковых идентификаторов группы (идентификатор Си). Для каждого отдельного параметра устанавливается максимальная погрешность измерений (в абсолютной или относительной величине), а также “время жизни” - интервал времени, в течение которого значение параметра считается валидным.

При получении нового значения параметра, включённого в группу, происходят следующие действия:

1. Значение параметра и время получения этого значения вносится в таблицу значений группы;

2. Определяются все “живые” значения параметров (те значения, для времени получения которых верно: $t_{now} \leq t_{recv} + timeout$;

3. Вычисляются абсолютные значения погрешностей для каждого параметра группы;

4. Строится множество отрезков, заданных центральной точкой (значение параметра) и радиусом (величина погрешности);

5. Строится пересечение полученных отрезков. Если пересечение отрезков пусто - ограничение нарушено.

Параметры ограничения:

- group - идентификатор группы связанных параметров (строка - идентификатор Си);
- measureError - допустимая ошибка измерения параметра. Может быть записана в абсолютной величине (без суффикса) или относительной величине (с суффиксом ’%’);
- timeout - время жизни последнего полученного значения.

Описание ограничения в файле протокола:

```
<signal identifier="signal1">
    <restrict type="bind" level="error">
        <param name="groupId" value="group1" />
        <param name="measureError" value="5%" />
        <param name="timeout" value="10s" />
    </restrict>
</signal>

<signal identifier="signal2">
    <restrict type="bind" level="error">
        <param name="groupId" value="group1" />
        <param name="measureError" value="0.3" />
        <param name="timeout" value="1s" />
    </restrict>
</signal>
```

Листинг 11: Связанные параметры

В случае несовпадения параметра level у описаний одной и той же группы связанных параметров, пользователь получит предупреждение, при этом будет выбрано самое сильное значение параметра.

2.6.7 Автоинкремент значения параметра

Автоинкремент - свойство целочисленного значения параметра увеличиваться на 1 с определённой частотой. Такие параметры могут использоваться для проверки работоспособности модуля абонента.

Проверяется соблюдение свойства автоинкремента значения параметра.

Параметры ограничения:

- timeout - длина максимального временного интервала, в течение которого значение параметра может оставаться неизменным.

Описание ограничения в файле протокола:

```
<signal identifier="signal1">
    <restrict type="autoincrement" level="error">
        <param name="timeout" value="1s" />
    </restrict>
</signal>
```

Листинг 12: Автоинкремент

3 Структура разработанного решения

Решение задачи было разработано с учётом архитектуры актуальной версии инструмента Opermon и программных компонент, реализующих его функциональность.

Для удобства реализации решение разрабатывалось с учётом возможностей инструментария Qt версии 4 и его библиотек для языка C++.

3.1 Внутреннее представление конфигурации анализатора

Пользователь передаёт анализатору данные о накладываемых ограничениях в составе файла описания протокола. При загрузке этого описания происходит преобразование данных во *внутреннее представление*, более пригодное для хранения в ОЗУ и с возможностью быстрого удобного доступа к отдельным элементам описания. При этом, внутреннее представление не содержит никакой информации, которую нельзя получить только из файла описания (взаимно однозначное соответствие описания и его внутреннего представления).

Внутреннее представление описания конфигурации анализатора - это пара массивов A_{exch}, A_{param}, содержащих пары *(идентификатор_объекта, список_ограничений)*. Здесь идентификатор объекта - набор признаков конкретного типа обмена или параметра, по которым объект однозначно определяется в системе. Массив A_{exch} содержит описания ограничений для обменов, A_{param} - для параметров.

Список ограничений - это массив, содержащий внутренние представления описаний ограничений, накладываемых на конкретный обмен или параметр.

Внутреннее представление отдельного ограничения - это структрура данных, которая имеет следующий набор полей данных:

- тип ограничения - один из элементов множества допустимых ограничений для заданного типа объекта;
- уровень критичности нарушения - элемент из множества уровней критичности нарушения ограничения (Info, Notice, Warning, Error, Fatal);
- значение ограничения - поле, хранящее значение атрибута value для данного ограничения;

- список дополнительных параметров - набор пар *(имя_параметра, значение_параметра)*. Набор допустимых имён и значений конкретных параметров задан отдельно для каждого типа ограничения.

Необходимость использования общего формата описания ограничений диктуется требованиями простоты и расширяемости: у программиста не должно возникнуть необходимости в редактировании средства загрузки параметров при добавлении новых типов ограничений.

При преобразовании данных из текстового представления во внутреннее представление значения неперечисляемых типов должны сохраняться в поле специального типа, допускающего последующее преобразование к различным машинным типам данных (как минимум, к целочисленным, строковым и значениям с плавающей точкой). Это диктуется требованием к расширяемости, так как обработчики конкретных ограничений должны иметь возможность использовать данные из описания конфигурации любым необходимым образом. В библиотеке инструментария Qt для хранения таких данных предложен специальный тип QVariant.

3.2 Получение требуемых данных

В данном разделе речь пойдёт о способах получения новых зарегистрированных обменов и значений параметров в рамках архитектуры актуальной версии Анализатора МКИО, а конкретно - в рамках архитектуры компонента tabexchange, реализующего функционал, необходимый для обработки и отображения обменов и параметров.

3.2.1 Получение новых обменов

В tabexchange есть множество объектов, получающих уведомления при регистрации новых обменов. Соответственно, для получения анализатором информации о новых обменах следует создать такой объект, который будет передавать информацию о новых обменах непосредственно анализатору.

В случае с новыми обменами, уведомления рассылаются с помощью вызова виртуального метода addExchange() у каждого из таких объектов при регистрации нового обмена.

3.2.2 Получение новых значений параметров

В актуальной версии tabexchange не было реализовано функционала для рассылки уведомлений о получении новых значений параметров.

Значения параметров запрашивались порциями по сигналу от внешних таймеров. Этого функционала вполне достаточно для реализации отображения значения параметра в соответствующей вкладке, а также для построения графиков зависимости значения параметра от времени.

Тем не менее, для анализатора требований требуется получение информации о новых значениях параметра сразу после получения этого значения из отдельного обмена. Поэтому в рамках данной работы был немного изменён и дополнен класс, реализующий хранение и подсчёт значений параметров при регистрации нового обмена (с полным сохранением уже существовавшего функционала).

В класс в дополнение к существующим полям был добавлен фильтр оперативно наблюдаемых параметров - массив, содержащий идентификаторы параметров, при изменении значений которых требуется рассылка уведомлений.

При получении нового значения параметра из этого списка, происходит рассылка уведомления всем объектам-наблюдателям. В данном случае уведомления рассылаются с использованием механизма “сигнал-слот”, реализованного в инструментарии Qt [2].

3.3 Обработка данных

3.4 Передача результатов проверки

Таблица 1: Параметры, передаваемые генератором

№ п/п	Отправитель	Получатель	Длина	Формат	Первое слово	Размер

4 Результаты апробации решения

4.1 Для генерируемых данных

Для проверки общей работоспособности решения без доступа к реальным или смоделированным ВС полезно провести апробацию средства на данных, полученных с помощью генератора трасс обменов.

Такой генератор был разработан ранее для демонстрации работоспособности инструмента Opermon. Для передачи данных агенту Opermon используется специальное виртуальное устройство-петля. С помощью генератора можно получить последовательность обменов, подходящую для проверки возможностей анализатора.

Во время работы генератор непрерывно передаёт набор сообщений определённых типов и содержания с некоторой частотой. Полный перечень параметров, получаемых от генератора,

5 Перспевктивы развития

Данная работа имеет следующие перспективы для возможного развития:

1. Усовершенствование интеграции с инструментами средства Opermon:
    - установка связей между строками отчёта анализатора и списком обменов для выделения в пользовательском интерфейсе обменов с обнаруженными проблемами;
    - разработка пользовательского интерфейса для редактирования требований к обменам, минуя подготовку файла протокола.
2. Усовершенствование пользовательского интерфейса вкладки с отчётом анализатора:
    - добавление дополнительных колонок данных: “Тип ограничения”, “Канал” и т.п.;
    - возможность сортировки по колонкам в таблице с отчётом анализатора;
    - вохможность фильтровать элементы отчёта по содержанию.
3. Расширение сферы применения разработанного средства на другие каналы информационного обмена.

6 Список литературы

1. Государственный стандарт РФ “Интерфейс магистральный последовательный системы электронных модулей” ГОСТ Р 52070-2003

2. Бланшет Ж. Qt 4: программирование GUI на C++, второе издание. КУДИЦ-Пресс, 2008. 736 с.