



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

<b>1</b>	<b>Введение</b>	<b>4</b>
1.1	Актуальность задачи . . . . .	4
<b>2</b>	<b>Введение</b>	<b>5</b>
2.1	Неформальная постановка задачи . . . . .	5
2.2	Определения и обозначения . . . . .	6
2.3	Формальная постановка задачи . . . . .	8
2.4	Уточнение исходных данных . . . . .	8
2.5	Формат описания входных данных . . . . .	10
2.6	Формирование проверяющих объектов . . . . .	12
2.7	Ограничения для сообщений . . . . .	13
2.8	Ограничения для параметров . . . . .	15
<b>3</b>	<b>Результаты апробации решения</b>	<b>21</b>
3.1	Для генерируемых данных . . . . .	21
<b>4</b>	<b>Перспективы развития</b>	<b>22</b>
<b>5</b>	<b>Список литературы</b>	<b>23</b>

## 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 Oregmon, 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 соответствующие 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 сравнивая 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 Введение

### 2.1 Неформальная постановка задачи

Целью работы является разработка методов и программных средств для анализа трасс данных обменов по МКИО с учётом заранее определённых требований. Примеры возможных требований:

- асинхронность сообщения, либо частота появления сообщения;
- соответствие контрольных сумм;
- гладкость передаваемого значения (ограничение на скорость изменения значения);
- автоматическое увеличение (автоинкремент) передаваемого значения во времени

Трасса состоит из слов (?) с признаками КС, ОС или СД. Из последовательности слов согласно одному из 10 шаблонов, приведённых в ГОСТ Р 52070-2003, составляются обмены, для которых выделяется набор полей:

- время начала обмена;
- продолжительность обмена;
- формат обмена - номер шаблона из ГОСТ Р 52070-2003
- адрес и подадрес абонента-отправителя и абонента-получателя (отдельно определяется фиктивный адрес КШ, так как он может не иметь реального адреса на шине);
- количество слов данных, переданных в рамках обмена;
- признак и тип ошибки, произошедшей при выполнении обмена;
- переданные данные.

Анализатору сообщается набор требований к конкретным сообщениям. Допускается также, что МКИО может функционировать в нескольких различных режимах, при этом для каждого режима формируется свой набор требований и условия переходов из одного режима в другой.

## 2.2 Определения и обозначения

**Обмен** -  $(M = \langle t, d, f, a_{src}, a_{dst}, s_{src}, s_{dst}, sz, e, p \rangle)$  - атомарный завершённый обмен информацией между абонентами канала МКИО, в соответствии с протоколом обмена по ГОСТ Р 52070-2003 [1]. Понятие обмена соответствует термину *сообщение* ГОСТ. С точки зрения анализа последовательностей обменов, каждый обмен характеризуется следующими атрибутами:

- $t$  - время начала обмена (мкс);
- $d$  - продолжительность обмена (мкс);
- $f$  - формат (тип) обмена по ГОСТ Р 52070-2003 - целое число в интервале  $[1, 10]$ ;
- $a_{src}, a_{dst}$  - адреса абонента-отправителя и абонента-получателя данных в рамках обмена - целые числа на отрезке  $[0, 32]$  ( $[0, 31]$  - допустимое множество адресов для МКИО, 32 - фиктивный адрес КШ (контроллера шины);
- $s_{src}, s_{dst}$  - подадреса абонента-отправителя и абонента-получателя данных в рамках обмена - целые числа на отрезке  $[1, 31]$ ;
- $sz$  - количество слов данных, передаваемых в сообщении - целое число на отрезке  $[0, 31]$ ;
- $e$  - признаки обмена - подмножество множества  $E$ , соответствующего набору признаков, передаваемых в ОС:
  - $Err$  - “Ошибка в сообщении” - признак достоверности принятых данных;
  - $Mntn$  - “Запрос на обслуживание” - требование ОУ или связанного с ним абонента (абонентов) на обслуживание;
  - $Grp$  - “Принята групповая команда” - указывает на приём ОУ достоверной групповой команды;
  - $Busy$  - “Абонент занят” - состояние занятости ОУ или интерфейса абонента, которое может повлиять на обмен информацией по каналу;
  - $Aflt$  - “Неисправность абонента” - техническое состояние абонента, связанного с ОУ;
  - $Eflt$  - “Неисправность ОУ” - техническое состояние ОУ;

- $p$  - полезная нагрузка (переданные данные в СД) - последовательность слов длины  $M_s z$ .

Атрибуты сообщения  $M$  обозначаются так:  $M_t, M_d, \dots, M_{a_{src}}, \dots, M_p$ . Дополнительно используются обозначения границ интервалов времени  $M_{start} = M_t, M_{end} = M_{start} + M_d$  и обозначение продолжительности обмена  $|M| = M_d$ . Если в названии сообщения используются индексы (например,  $M^i$ ), то атрибуты обозначаются так:  $M_t^i, M_d^i, \dots, M_{a_{src}}^i, M_p^i$ .

Дополнительно в сообщении  $M$  может передаваться его контрольная сумма, значение которой будем в дальнейшем обозначать  $M_{ch}$ .

**Экземпляр обмена** - совокупность  $M$  значений всех атрибутов обмена. Экземпляр обмена характеризуется типом  $Type(M)$ , временем начала  $M_t$ , продолжительностью  $M_d$ , признаками  $M_e$  и полезной нагрузкой  $M_p$ .

**Тип обмена** ( $T = Type(M)$ ) - совокупность значений атрибута обмена  $\langle f, a_{src}, a_{dst}, s_{src}, s_{dst}, sz \rangle$ . Каждый экземпляр обмена, таким образом, соответствует одному и только одному типу данных.

**Трасса обменов**  $T = \{M^1, M^2, \dots, M^n\} : M_{end}^i \leq M_{start}^{i+1} \forall i = \overline{1, n-1}$  - конечная последовательность обменов. Время начала обменов  $M_t^i$  отсчитывается относительно момента начала записи трассы обмена.

**Описание цепочки обменов**  $D$  - конечный автомат, множеством состояний и входным алфавитом которого является подмножество множества типов обмена. Определяя это множество, а также начальное состояние, множество конечных состояний и функцию перехода, пользователь определяет логически связанные последовательности обменов в трассе.

**Цепочка обменов**  $C$  - логически связанная последовательность обменов в трассе (не обязательно непосредственно следующих друг за другом), определяемая описанием цепочки  $D$ .

В дальнейшем будем пользоваться следующим обозначением: описанию цепочки  $D_i$  в трассе  $T$  соответствует последовательность цепочек  $C_i = \{C_i^1, \dots, C_i^k\}$ .

**Функция корректности последовательности цепочек**  $R$  - функция  $R : C^1, \dots, C^q \rightarrow \{1, 0\}$ , определяющее корректность входной последовательности цепочек, т. е.  $R(C) = 1 \Leftrightarrow C$  - корректная последовательность цепочек. Требования корректности определяются пользователем для проверки непосредственно передаваемых данных, в том числе с учётом типов, порядка и времени их передачи.

## 2.3 Формальная постановка задачи

**Исходные данные:** зарегистрированная трасса обменов на канале МКИО  $T = \{M^1, \dots, M^n\}$ , множество описаний цепочек  $D = \{D_1, \dots, D_m\}$  и функция корректности последовательностей цепочек  $R$ .

**Требуется:** разработать набор алгоритмов и реализовать программные средства, обеспечивающие решение следующих задач:

- поиск цепочек  $C_i$  в трассе  $T$  согласно описаниям цепочек  $D_i$ ;
- вычисление статистических характеристик цепочек обменов  $C_i$ :
  - частота появления цепочек  $f_i$ ;
- проверка корректности последовательностей цепочек  $C_i$  с помощью функции  $R$ .

## 2.4 Уточнение исходных данных

### 2.4.1 Режимы работы системы

Множество описаний цепочек делится дополнительно на подмножества, образующие *режимы* работы анализируемой системы. Дополнительно к каждому описанию цепочки дописывается поле, содержащее новое значение режима работы (в том случае, если появление данной цепочки это подразумевает).

### 2.4.2 Функция корректности последовательности цепочки

Для упрощения задачи полезно уточнить определение функции корректности последовательности так, чтобы она требовала меньше входных данных с сохранением всех свойств.

Введём дополнительно определение **подцепочки**  $\hat{C}_k^t$  цепочки  $C = \{M^1, \dots, M^n\}$  длительности  $t$  как максимальной подпоследовательности последовательности  $\{M^j\}_{k+p}^{j=k}$ , где время между началом первого и последнего сообщения не превышает  $t$ :  $M_{start}^{k+p} - M_{start}^k \leq t$ . Требование максимальной длительности необходимо для попадания в подцепочку всех обменов на отрезке времени  $[M_{start}^k, M_{start}^k + t]$ .

Считается, что функция проверки корректности  $R$  может быть одной из нижеперечисленных, определённых на соответствующих множествах обменов, имеющих требуемые в формулировках атрибуты:



- $R_p(M^i, M^{i+1}, f_{Type(M)})$  - функция, проверяющая асинхронность сообщения, либо соответствие реальной частоты появления обмена  $M$  заранее заданной  $f_{Type(M)}$ :  $R_p(M^i, M^{i+1}, f_{Type(M)}) = 1 \Leftrightarrow (f_{Type(M)} = 0) \vee (M_{start}^{i+1} - M_{start}^i \geq f_{Type(M)}^{-1}, f_{Type(M)} > 0)$ .  $R_p$  определена на множестве всех обменов;
- $R_c(M^i)$  - функция, устанавливающая соответствие подсчитанной контрольной суммы тела обмена и переданной в обмене:  $R_c(M) = 1 \Leftrightarrow checksum(M^i) = M_{ch}^i$ .  $R_c$  определена на множестве обменов, имеющих атрибут  $M_{ch}$ ;
- $R_{ai}(\hat{C}_{t_{Type(M)}}^i)$  - функция, определяющая автоматическое увеличение (автоинкремент) передаваемого значения во времени не реже, чем каждый интервал времени длительности  $t_{Type(M)}$ :  $R_{ai}(\hat{C}_{t_{Type(M)}}^i) = 1 \Leftrightarrow (M_{start}^{i+p} - M_{start}^i > t_{Type(M)}) \wedge (M_{ai}^{i+p} - M_{ai}^i > 0)$ , где  $p$  - длина подцепочки  $\hat{C}^i$ ,  $M_v^j$  - заранее определённый увеличиваемый атрибут обмена.  $R_{ai}$  определена на множестве обменов, имеющих атрибут  $M_{ai}$ .

Константы  $f_{Type(M)}$  и  $t_{Type(M)}$  можно занести внутрь функций  $R$ . Также стоит расширить функции  $R_c$  и  $R_{ai}$  на множество всех возможных обменов, доопределив их значением 1. Таким образом, мы получаем функции  $R_p^{Type(M)}(M^i, M^{i+1})$ ,  $R_c^{Type(M)}(M)$ ,  $R_{ai}^{Type(M)}(M)$

Наличие атрибутов  $M_{ch}$  и  $M_{ai}$  определяется типом сообщения  $M$ .

Таким образом, в общем случае функция обмена существенно зависит от:

- типа обрабатываемого обмена  $Type(M)$ ;
- одного обмена, или двух обменов, или подцепочки обменов длительности  $t$ ,

что упрощает описание множества определения функции  $R$ .

## 2.5 Формат описания входных данных

Формат входных данных обратно совместим с форматом, использованным в некоторых версиях Орегмон 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>
  <!-- ... -->
</piv>
```

```

</abonents>

<restricts>
  <!-- restricts -->
  <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` - идентификатор 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 ПИБ).

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 - идентификатор 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 - идентификатор 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).

## 2.6 Формирование проверяющих объектов

## 2.7 Ограничения для сообщений

### 2.7.1 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

### 2.7.2 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

### 2.7.3 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 строковый 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.

#### 2.7.4 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

## 2.8 Ограничения для параметров

### 2.8.1 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

### 2.8.2 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



### 2.8.3 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

### 2.8.4 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

### 2.8.5 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

### 2.8.6 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.

### 2.8.7 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

Таблица 1: Параметры, передаваемые генератором						
№ п/п	Отправитель	Получатель	Длина	Формат	Первое слово	Размер

### 3 Результаты апробации решения

#### 3.1 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 Oregmon. Для передачи данных агенту Oregmon 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,

## 4 Перспективы развития

Данная работа имеет следующие перспективы для возможного развития:

1. Усовершенствование интеграции с инструментами средства Opremon:
  - 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;
  - возможность фильтровать элементы отчёта по содержанию.
3. Расширение сферы применения разработанного средства на другие каналы информационного обмена.

## 5 Список литературы

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