

Содержание

1	Введение	2
1.1	Неформальная постановка задачи	2
1.2	Определения и обозначения	3
1.3	Формальная постановка задачи	5
1.4	Уточнение исходных данных	5
1.5	Формат описания входных данных	7
1.6	Формирование проверяющих объектов	9
2	Список литературы	10

1 Введение

1.1 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.

1.2 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.

1.3 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 .

1.4 Уточнение исходных данных

1.4.1 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).

1.4.2 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 .

1.5 Формат описания входных данных

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

Формат входных данных основан на XML.

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

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

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

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

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

```

<restricts>
  <!-- restricts -->
  <restrict type="" level="">
    <!-- special restriction params -->
  </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`).

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

- `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`).

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-1553В 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-1553В для оконечных устройств не указан адрес ОУ. 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 совпадают.

1.6 Формирование проверяющих объектов

2 Список литературы

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