

УДК 629.78.018

И.Б. ТУРКИН, Е.В. СОКОЛОВА

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

МОДЕЛЬ ВЫЧИСЛЕНИЙ, УПРАВЛЯЕМЫХ ДАННЫМИ, В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Показано, что контроль точности, достоверности и актуальности вычислений в программном обеспечении систем реального времени (ПО СРВ) является актуальной задачей. Предложено в качестве теоретического решения данной задачи использовать модель вычислений, управляемых данными, с учетом неопределенностей: временной, связанной с неизвестным точным временем их получения, и интервальной, вызванной ошибками измерений и преобразований. Показано, что в существующих паттернах-шаблонах проектирования ПО, например, в «Шине данных», возможен учет и обработка этих неопределенностей. Сформулированы основные правила преобразования информации, обеспечивающие автоматический контроль и верификацию данных. В качестве средства исследования и моделирования вычислителей предложено использовать сеть Петри.

Ключевые слова: программное обеспечение систем реального времени, модель вычислений, потоковые сети Петри, шаблоны объектно-ориентированного программирования.

Введение

Системы реального времени (СРВ), в отличие от обычных информационных систем, должны обеспечить гарантированную реакцию на входное воздействие в известное, заранее специфицированное время [1]. Ответ, полученный поздно, приравнивается к неправильному – «The right answer late is wrong». По степени важности последствий несоблюдения ограничений на время реакции выделяют две группы систем: «мягкого» и «жесткого» реального времени. В первом случае несоблюдение требований реального времени не является катастрофическим по отношению к цели работы СРВ, но ухудшает показатели качества всей информационной системы. Во втором случае, несоблюдение требований реального времени приводит к невозможности выполнения целевой функции системы.

В программном обеспечении систем реального времени (ПО СРВ) точность и достоверность данных являются критическими факторами, во многом определяющими безопасность и эффективность функционирования системы в целом. Наряду с непосредственно измеряемыми параметрами в таких системах широко используются вычисляемые величины, описывающие недоступные для прямых измерений характеристики состояния объекта исследования.

1. Обзор литературы

В настоящее время в практике объектно-ориентированного проектирования широко приме-

няются паттерны (шаблоны) [2], позволяющие поддерживать согласованное состояние взаимосвязанных объектов (данных).

Шаблон «Наблюдатель» определяет зависимость типа «один ко многим» между объектами таким образом, что при изменении состояния одного объекта все зависящие от него оповещаются об этом событии. При реализации шаблона «наблюдатель» обычно используются классы (рис. 1).

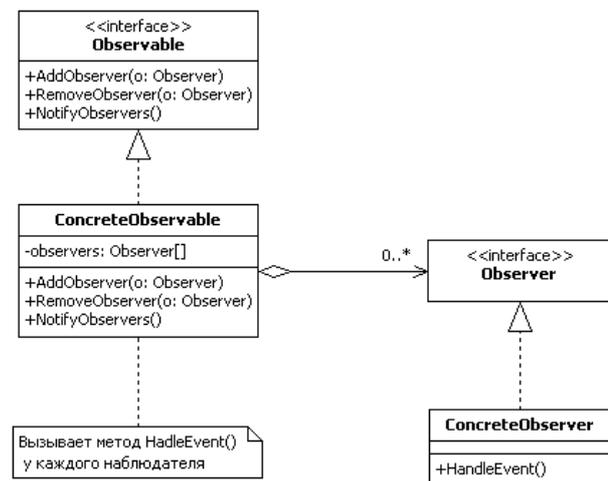


Рис. 1. Структура шаблона «Наблюдатель»

– Observable – интерфейс, определяющий методы для добавления, удаления и оповещения наблюдателей (подписчиков).

– Observer – интерфейс, с помощью которого наблюдаемый объект (издатель) оповещает наблюдателей.

- ConcreteObservable – конкретный класс, который реализует интерфейс Observable.
- ConcreteObserver – конкретный класс, который реализует интерфейс Observer.

Шаблон «Шина данных» наследуется от шаблона «Наблюдатель» и определяет зависимость типа «многие ко многим» между объектами таким образом, что при изменении состояния многих объектов все зависящие от него оповещаются об этом событии. Шаблон использует центральное хранилище информации, полученной разными способами, что позволяет хранить исходные данные совместно с результирующими [3], а любой «подписчик» осуществляет взаимоотображение области данных.

Описанные шаблоны использовались при разработке ПО для испытаний бортовых систем космического аппарата [4] на уровне сервисов поддержки логики программы. Хранилище данных (таблица переменные/подписчики) обеспечивает логически непротиворечивую и надежную работу алгоритмов, состав которых может изменяться во время исполнения основной программы. Наблюдаемые объекты данных обеспечивают хранение информации о состоянии внешних (физических) сигналов, внутренних переменных, необходимых для синхронизации основных программных компонентов. Передача команд управления во внешнюю среду также производится после изменения соответствующих объектов данных в данной таблице. Подписчики отвечают за логическую обработку информации, их работа инициируется изменением объектов данных в хранилище данных.

Именно на основе паттерна «Шина данных» в программе решена проблема разбиения системы на относительно простые, параллельно работающие, независимые компоненты, передача информации между ними при этом осуществляется по единому механизму.

Метод вычислений, при котором выполнение каждой операции производится при готовности всех ее операндов, называется data flow (вычисления, управляемые потоком данных), при таком методе последовательность выполнения команд заранее не задается; используется также понятие Data Driven Computing (вычисления, управляемые данными) [5]. В системе data flow вместо императивного указания выполнить некоторую команду используется разрешение ее выполнить, при этом схема data flow противопоставляется традиционной схеме с потоком управления (control flow).

Модель вычислений, управляемых данными, оказывается весьма эффективной в ПО СРВ, которое активно взаимодействует с внешней средой и при этом должно гарантировать абсолютную корректность результата в условиях ограниченного

времени, за которое он должен быть получен.

Наличие множества «подписчиков», каждый из которых отвечает за решение своей задачи, позволяет естественным образом контролировать ограничения на время решения данных задач. В зависимости от присущих им временных ограничений все задачи могут быть классифицированы на следующие виды:

- задачи, в которых временные ограничения настолько слабы, что ими можно просто пренебречь;

- задачи «мягкого» реального времени (soft real time task), для которых достаточно обеспечить выполнение временных ограничений в среднем, при этом допустимы отдельные нарушения таких ограничений;

- в задачах «жесткого» реального времени (hard real time task) превышение времени решения может привести к катастрофическим последствиям.

Задачи-преобразователи разделяют на [6]:

- моноэлементные, когда один входной сигнал порождает один выходной;

- полиэлементные, когда один выходной сигнал порождается совокупностью входных, то есть подписчик вычисляет агрегированные значения по некоторому множеству входных переменных.

Применение паттерна «Шина данных» в последнем случае требует доработки для применения в системах реального времени, так как перерасчет результата происходит при изменении любой входной переменной. В случае, когда произошло изменение только части переменных из этого множества, выполнение промежуточного расчета с использованием части новых и части устаревших объектов данных избыточно, и, скорее всего, ошибочно. При этом заранее не всегда известно, поступление какого именно сигнала завершает обработку всего входного информационного пакета, то есть невозможно предсказать, последует ли за поступлением нового значения одного из сигналов новое значение другого. Следовательно, необходимо отслеживать режим поступления обновлений объектов данных.

Реализация такого отслеживания с одновременным контролем указанных выше ограничений требует учета временных свойств объектов данных при инициировании работы «подписчиков», осуществляющих преобразование этих объектов. В статье [7] предложено решение, когда для всех задач-преобразователей информации предполагаются известными ограничения на:

- абсолютную корректность входных данных, когда выполнение задачи разрешается, если все входные данные имеют временную отметку получения, соответствующие требованиям решаемой задачи;

- относительную корректность входных данных, характеризующие допустимые временные расхождения между объектами данных.

Учет таких временных ограничений позволяет автоматически контролировать корректность и своевременность проводимых вычислений в подписчиках, решающих задачи «жесткого» реального времени по критериям актуальности исходных данных.

Пример. Рассмотрим вычисление некоторой функции $f(\{x_i\}), i \in 1..k$. Предположим, что временем процессорной обработки информации можно пренебречь, а собственно исходные данные $\{x_i\}$ измеряются с периодом T тремя способами:

- параллельно, когда в результате одновременных запросов к разным измерительным устройствам обновляется все множество необходимых данных;
- последовательно, когда это же обновление происходит путем выполнения нескольких запросов к одному измерительному устройству;
- в результате выполнения пакетного запроса, обновляющего все множество данных.

Учет неопределенности порождения исходных данных позволяет контролировать абсолютную корректность результата вычислений (рис. 2).

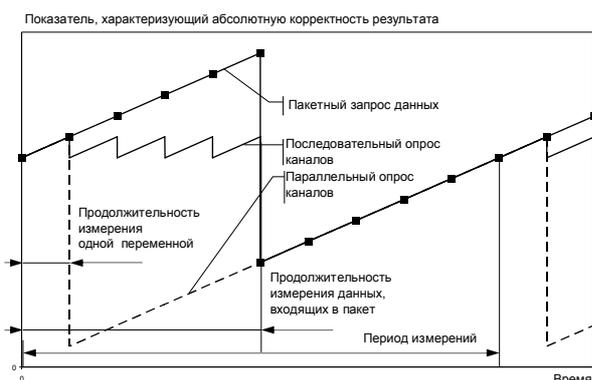


Рис. 2. Влияние способа организации измерений на абсолютную корректность результата вычислений

Помимо временных ограничений в практических задачах необходимо осуществлять допусковый контроль точности проводимых преобразований, что позволит реализовать автоматическую валидацию выполняемых расчетов. Контроль точности при этом следует вести с учетом рекомендаций международной организации по стандартизации [8], которые направлены на замену модели интерпретации измерений, в основе которой – погрешность измерения, на модель неопределенности. В настоящее время эти рекомендации используются в испытательных лабораториях [9] и плохо поддерживаются имеющимися инструментальными средствами.

В общем случае точность интервального результата полностью определяется [10]:

- неопределенностью в задании исходных данных;

- округлениями при выполнении операций, изменяющих или порождающих интервальные объекты;

- приближенным характером используемого численного метода;

- степенью учета зависимостей между участвующими в вычислении интервальными объектами (переменными и константами).

Увеличение точности расчетов (уменьшение ширины результирующего интервала) достигается за счет компенсации влияния этих факторов.

Степень близости множества значений функции и ее интервальной оценки зависит от выбора способа расчета функции.

Например, пусть $f(x) = x - x^2, X = [0, 1]$. Тогда, различные аналитические выражения для f дают различные результаты:

$$f^{(0)}(x) = x - x^2 \Rightarrow f^{(0)}(x) = [-1, 1],$$

$$f^{(1)}(x) = x(1 - x) \Rightarrow f^{(1)}(x) = [0, 1],$$

$$f^{(2)}(x) = \frac{1}{4} - \left(x - \frac{1}{2}\right)\left(x - \frac{1}{2}\right) \Rightarrow f^{(2)}(x) = \left[0, \frac{1}{4}\right],$$

$$f^{(3)}(x) = \frac{1}{4} - \left(x - \frac{1}{2}\right)^2 \Rightarrow f^{(3)}(x) = \left[0, \frac{1}{4}\right].$$

Применив методы функционального анализа, получим истинный диапазон изменения функции f :

$$W(f, [0, 1]) = \{x - x^2 | 0 \leq x \leq 1\} = \left[0, \frac{1}{4}\right].$$

Задача получения для данного множества машинно-представимых чисел самого узкого интервала, содержащего объединенное расширение соответствующей функции, может ставиться как оптимизационная. В этой связи актуальным является создание модели представления объектов данных и методов их обработки для ПО СРВ, которые бы обеспечивали контроль точности, достоверности и актуальности вычисляемых параметров.

2. Постановка задачи

Проведенный обзор публикаций показал, что широкое распространение получили паттерны проектирования ПО, которые ориентированы на реализацию модели вычислений, управляемых данными. Вместе с тем, в литературе отсутствуют теоретические проработки и практические рекомендации по обеспечению автоматической валидации выполняемых вычислений, что особенно актуально для ПО СРВ, в которых необходим комплексный учет неопределенности результата измерений, времени его получения, а задача получения самого узкого интервала вычисляемых параметров может ставиться как оптимизационная.

Целью работы является теоретическое обоснование модели вычислений, управляемых данными, в ПО СРВ, которая позволит обеспечить контроль точности, достоверности и актуальности вычисляемых параметров.

3. Результаты исследований

3.1 Модель представления данных

Определение. Объект данных (ОД) – это формализованное представление измеренной или вычисленной величины с учетом погрешностей измерения и преобразований, а также времени, когда эта величина получена:

$$\text{Data}_i = \langle \text{Value}_i, T, k \rangle, \quad (1)$$

где Value_i – интервальное значение i -го объекта данных в некоторый интервал времени

$T = \{t \mid t_{\text{query}} \leq t \leq t_{\text{answer}} \mid t > 0\}$ от момента формирования запроса до момента получения ответа,

$$\begin{aligned} \text{Value}_i &= [\underline{\text{Value}}_i, \overline{\text{Value}}_i] = \\ &= \{ \text{Value}_i \in \mathbb{R} \mid \underline{\text{Value}}_i \leq \text{Value}_i \leq \overline{\text{Value}}_i \}, \end{aligned}$$

а операции над ними производятся согласно правилу:

$$\begin{aligned} \text{Value}_A \otimes \text{Value}_B &= \\ &= \left\{ \text{value} \in \mathbb{R} \mid \begin{array}{l} \text{value} = \text{value}_1 \otimes \text{value}_2, \\ \text{value}_1 \in \text{Value}_A, \text{value}_2 \in \text{Value}_B \end{array} \right\}, \\ \otimes &\in \{+, -, \cdot, / \} \end{aligned}$$

k – динамическая характеристика для описания максимальной скорости изменения величины, описываемой кортежем за период T .

В рассмотрении вводится именно интервал времени, поскольку в результате пакетного, наиболее часто применяемого запроса на ввод информации возвращается пакет данных, сопровождаемый отметкой времени всего пакета, а не отдельных данных, в него входящих.

3.2 Правила выполнения операций над ОД

Рассмотрим правила выполнения арифметических операций над ОД. Множество всех ОД обозначим символом O . Пусть A и B – ОД с совпадающими интервалами времени:

$$A \in O, B \in O, T_A = T_B = T_0.$$

Результатом выполнения арифметической операции над ОД A и B будет также ОД, полученный по правилам:

$$A + B = \langle \text{Value}_A + \text{Value}_B, T_0, k_A + k_B \rangle, \quad (2)$$

$$A - B = \langle \text{Value}_A - \text{Value}_B, T_0, k_A + k_B \rangle, \quad (3)$$

$$\begin{aligned} A \cdot B &= \\ &= \left\langle \text{Value}_A \cdot \text{Value}_B, T_0, \right. \\ &= \left. \left\langle k_A \cdot |\text{Value}_B| + k_B \cdot |\text{Value}_A| \right\rangle \right\rangle, \quad (4) \end{aligned}$$

$$\begin{aligned} \frac{A}{B} &= \\ &= \left\langle \frac{\text{Value}_A}{\text{Value}_B}, T_0, \frac{k_A \cdot |\text{Value}_B| + k_B \cdot |\text{Value}_A|}{\rho^2(\text{Value}_B, 0)} \right\rangle, \quad (5) \\ &\text{если } 0 \notin \text{Value}_B, \end{aligned}$$

где амплитуда интервала

$$|\text{Value}| = \max \{ \underline{\text{Value}}, \overline{\text{Value}} \},$$

а расстояние между интервалами

$$\begin{aligned} \rho(\text{Value}_A, \text{Value}_B) &= \\ &= \max \left\{ \left| \underline{\text{Value}}_A - \underline{\text{Value}}_B \right|, \left| \overline{\text{Value}}_A - \overline{\text{Value}}_B \right| \right\}. \end{aligned}$$

В формулах (2) - (5) интервал значения ОД вычисляется согласно стандартным правилам интервальной математики. Интервал времени результата по очевидным соображениям устанавливается такой же, как и у обоих операндов. Правила определения динамической характеристики продиктованы физическим смыслом данного компонента ОД: представляя верхнюю оценку абсолютной величины производной искомого параметра, динамическая характеристика вычисляется по правилам, похожим на обычные правила нахождения производных. Учитывая интервальную природу значений ОД в формулах (4), (5), $|\text{Value}|$ и $\rho(\text{Value}, 0)$ используются таким образом, чтобы в итоге получить верхнюю оценку скорости изменения результата.

Для автоматической оценки влияния несинхронности получения данных на достоверность результата операции в модель введены три операции.

1. Приведение ОД к моменту времени τ . Результатом приведения ОД $A = \langle V_A, t, k_A \rangle$ к моменту времени τ будем называть ОД $A_\tau = \langle V_{A_\tau}, \tau, k_A \rangle$, где

$$\underline{\tau} = \tau = \overline{\tau}, \underline{V}_{A_\tau} = \overline{V}_A + k_A \cdot \max(\tau - \underline{t}, \bar{t} - \tau),$$

$$\overline{V}_{A_\tau} = \underline{V}_A - k_A \cdot \max(\tau - \underline{t}, \bar{t} - \tau).$$

2. Приведения ОД к расширенному временному интервалу. Под расширенным временным интервалом для ОД $A = \langle V_A, t, k_A \rangle$ будем понимать некоторый интервал времени t_e такой, что $\underline{t}_e \leq \underline{t}, \bar{t} \leq \overline{t}_e$. Тогда результатом приведения ОД $A = \langle V_A, t, k_A \rangle$ к временному интервалу t_e будем

называть ОД $A_e = \langle V_{A_e}, t_e, k_A \rangle$, где

$$\overline{V_{A_e}} = \overline{V_A} + k_A \cdot \max(\overline{t_e} - \underline{t}, \underline{t} - \overline{t_e}),$$

$$\underline{V_{A_e}} = \overline{V_A} - k_A \cdot \max(\overline{t_e} - \underline{t}, \underline{t} - \overline{t_e}).$$

3. Интерполяция ОД к моменту времени τ . Результатом интерполяции ОД $A = \langle V, t, k_A \rangle$ к моменту времени τ при известных значениях A_i, A_{i+1} , таких, что: $\underline{t}_i < \tau < \overline{t}_{i+1}$ будем называть ОД $A|_\tau = \langle V_A, t, k_A \rangle$, где

$$\overline{V_A}|_\tau = \min(\overline{V_{A_i}} + k_A (\tau - \underline{t}_i), \overline{V_{A_{i+1}}} + k_A (\overline{t}_{i+1} - \tau)),$$

$$\underline{V_A}|_\tau = \max(\underline{V_{A_i}} - k_A (\tau - \underline{t}_i), \underline{V_{A_{i+1}}} - k_A (\overline{t}_{i+1} - \tau)).$$

Операция приведения ОД к общему временному интервалу позволяет распространить арифметические операции (2) - (5) на случай произвольных временных интервалов ОД. Для этого достаточно потребовать предварительного приведения к общему временному интервалу, либо моменту времени все ОД, участвующие в вычислении какого-либо расчетного параметра.

Пример выполнения операций над объектами данных приведен на Рис. 3, на котором обозначены:

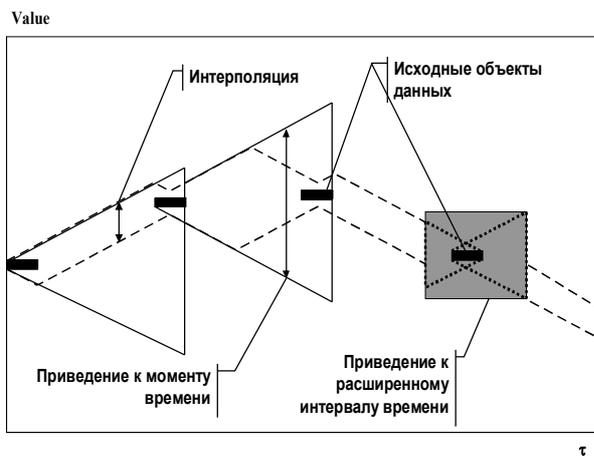


Рис. 3. Пример выполнения операций над ОД

- исходные объекты данных – черные прямоугольники;
- объект данных, полученный в результате приведения ОД к расширенному временному интервалу – серый прямоугольник;
- сплошная линия характеризует зависимость результата выполнения операции приведения ОД от времени;
- пунктирная линия характеризует зависимость результата выполнения операции интерполяции ОД от времени.

3.3 Моделирование вычислений, управляемых данными, с помощью сетей Петри

Модель вычислений, управляемых данными, отличается от общего случая, который реализует два типа отношений между преобразователями информации: связь по передаче управления, когда один преобразователь выполняется вслед за другим, и связь по передаче информации, когда преобразователь использует информацию, выработанную в результате работы другого преобразователя [11]. В модели вычислений, управляемых данными, только изменение данных инициирует работу последующих преобразователей.

Взаимоотображение данных, выполняемое преобразователями-подписчиками, представим графом:

$$N = \langle \text{Data}, \text{Task}, W \rangle, \quad (2)$$

где Data – непустое множество данных; Task – непустое множество задач-преобразователей; $F \subseteq \text{Data} \times \text{Task} \cup \text{Task} \times \text{Data}$ – отношения инцидентности (правила передачи данных).

Например, для сети (Рис. 4) места Data_1 и Data_2 входные для перехода Task_1 , а место Data_4 выходные.

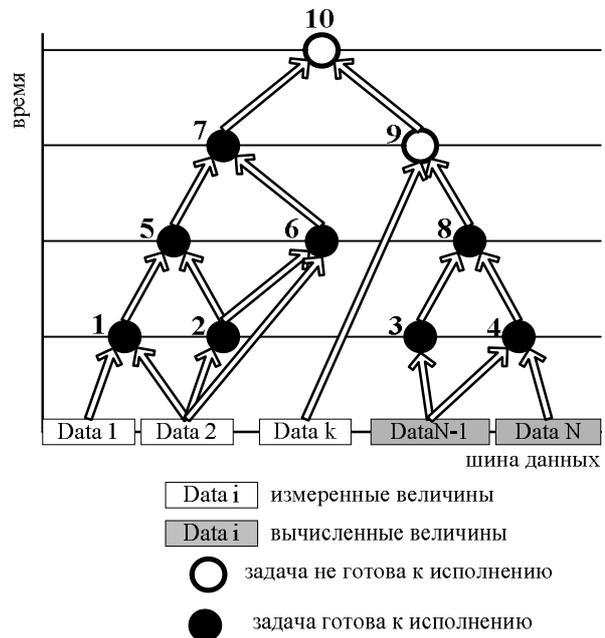


Рис. 4. Иерархия задач-преобразователей информации

При моделировании вычислительных процессов, управляемых данными, целесообразно использовать сети Петри и методы их функционального анализа [12]. Сеть Петри – это разновидность направленного графа, с двумя типами вершин: местами и переходами, соединенных между собой дугами.

ми-связями. Все места имеют первоначальную разметку определенным количеством фишек. В общем случае сеть Петри формально задается пятёркой:

$$N = (P, T, F, H, M_0),$$

где P – непустое конечное множество позиций (мест),

T – непустое конечное множество переходов,

F и H – матрицы инцидентности, причем

$F: P \times T$ – отображение множества P на T ,

$H: T \times P$ – отображение множества T на P ,

M_0 – начальная разметка (маркировка) – вектор, который содержит информацию о количестве меток (фишек, маркеров) в соответствующих позициях (местах) сети.

Потоковые сети Петри – сети, моделирующие потоковые системы, в которых осуществляется управление данными [13]. В потоковой сети Петри переходы интерпретируются как операторы или вычислительные функции, места интерпретируются как очереди, а данные – как фишки. Если переход имеет n входов, то он реализуется n -местной функцией, которая срабатывает сразу же при наличии фишек во всех входных местах. Для применения потоковых сетей Петри в ПО СРВ, введем следующие уточнения (Рис. 5).

Все объекты данных – токены, представлены в едином формате, который включает очередь, характеризующую предысторию – изменение состояния переменной в моменты, предшествующие обработке. Состояние объекта данных характеризует модель данных, представленную в 3.1. Токенам соответствуют места P в потоковой сети Петри.

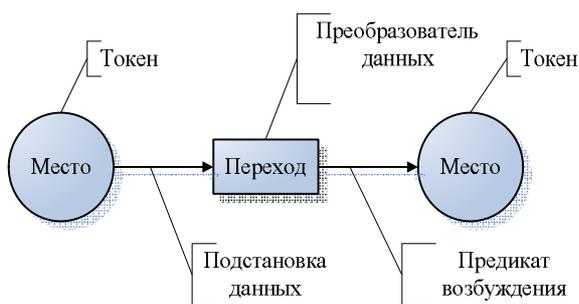


Рис. 5. Уточнение для применения потоковых сетей Петри в ПО СРВ

Переходы T связывают входные и выходные места и срабатывают при появлении хотя бы одной фишки (обновлении данных) во входном месте. Срабатывание переходов перемещает фишки из входных мест в выходные. Именно в переходах осуществляется преобразование входных данных подписчиками.

Дуги F , связывающие входные места и переходы, соответствуют операциям подготовки данных.

Дуги H , связывающие переходы и выходные места, реализуют операцию записи данных на шину данных при истинном значении предиката возбуждения, которое достигается, если вычисленное значение порождаемого объекта данных характеризуется погрешностью меньшей, чем у существующего.

Первоначальная разметка M_0 задается определенным количеством фишек, соответствующих поступлению данных от внешних измерительно-преобразовательных устройств.

Заключение

В программном обеспечении систем реального времени (ПО СРВ) точность и достоверность данных являются критическими факторами, во многом определяющими безопасность и эффективность функционирования системы в целом.

В качестве теоретического решения данной задачи определена модель вычислений, управляемых данными, которая учитывает два вида неопределенностей: временную, связанную с неизвестным точным временем их получения, и интервальную, вызванную ошибками измерений и преобразований.

Сформулированы основные правила преобразования информации, обеспечивающие автоматический контроль и верификацию данных.

В качестве средства исследования и моделирования вычислителей предложено использовать сеть Петри.

Литература

1. Ключев А.О. Программное обеспечение встроенных вычислительных систем / А.О. Ключев, П.В. Кустарев, Д.Р. Ковязина, Е.В. Петров. – СПб.: СПбГУ ИТМО, 2009. – 212 с.
2. Приемы объектно-ориентированного проектирования. Паттерны проектирования // Гамма Э. Хелм. Р., Джонсон Р., Дж. Влассидес. – СПб.: Питер, 2006. – 366 с.
3. Douglass B. Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems / Bruce Powel Douglass // – Addison Wesley, 2002. – 528с.
4. Компонентная модель программного обеспечения для испытаний бортовых систем космического аппарата / И. Туркин, П. Лучишев, И. Перекопский, И. Мосиенко // СТА: современные технологии автоматизации. – 2006 – № 1. – С. 66-73.
5. Параллельные вычисления: учеб. пособие / В.М. Баканов. – МГУПИ: Москва, 2006. – 124 с.
6. Литвинская О.С. Применение целевой функции в системе поддержки принятия решений при выборе варианта реализации алгоритма последовательной обработки данных / О.С. Литвинская,

И.И. Сальников // Искусственный интеллект. – 2009. – №3. – С. 115-120.

7. Кузнецова Ю. А. Адаптивное управление сервером данных в SCADA-системах / Ю.А. Кузнецова, Е.В. Соколова // Радиоэлектронні і комп'ютерні системи. – 2009 – № 7. – С. 26-31.

8. EA-4/02. Expression of the uncertainty of measurements in calibration, 1999.

9. ДСТУ ISO/IEC 17025-2001. Загальні вимоги до компетентності випробувальних та калібрувальних лабораторій.

10. Аленфельд Г. Введение в интервальные вычисления / Г. Аленфельд, Ю. Херцберге. – М.: Мир, 198. – 360 с.

11. Касьянов В.Н. Графы в программировании: обработка, визуализация и применение / В.Н. Касьянов, В.А. Евстигнеев. – СПб.: БХВ-Петербург, 2003. – 1104 с.

12. Ломазова И.А. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой / И.А. Ломазова. – М.: Научный мир, 2004. – 208 с.

13. Питерсон Дж. Теория сетей Петри и моделирование систем пер. с англ. / Дж. Питерсон. – М.: Мир, 1984. – 264 с.

Поступила в редакцию 15.02.2010

Рецензент: д.т.н., проф., зав. каф. економіко-математического моделирования В.М. Вартамян, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

МОДЕЛЬ ОБЧИСЛЕНЬ, ЯКІ КЕРОВАНІ ДАНИМИ, У ПРОГРАМНОМУ ЗАБЕЗПЕЧЕННІ СИСТЕМ РЕАЛЬНОГО ЧАСУ

І.Б. Туркін, Є.В. Соколова

Показано, що контроль точності, достовірності та актуальності обчислень у програмному забезпеченні систем реального часу (ПЗ СРЧ) є актуальною задачею. Як теоретичне розв'язання цієї задачі запропоновано використовувати модель обчислень, керованих даними, з обліком невизначеностей: за часом, зв'язаних з невідомим часом їх отримання, та інтервальної, пов'язаних з помилками вимірів та перетворень. Показано, що в існуючих паттернах-шаблонах проектування ПЗ, наприклад у «Шині даних», можливо обліковувати та обробляти ці невизначеності. Сформульовані основні правила перетворення інформації, що забезпечують автоматичний контроль та верифікацію даних. В якості засобу дослідження та моделювання обчислювачів запропоновано використовувати мережу Петрі.

Ключові слова: програмне забезпечення систем реального часу, модель обчислень, потокові мережі Петрі, шаблони об'єктно-орієнтованого програмування.

THE CALCULATIONS MODEL, OPERATED DATA, IN THE REAL TIME SYSTEMS SOFTWARE

I.B. Turkin, E.V. Sokolova

It is shown, that the control of accuracy, reliability and an relevance of calculations in the real time systems software is an actual problem. It is offered to use as the theoretical decision of the given problem model of the calculations operated data, with the account uncertainties: time, connected with an unknown exact times of their reception, and the interval, caused errors of measurements and transformations. It is shown, that in existing patterns-templates of designing SOFTWARE, for example in «the Tyre of data», the account and processing of these uncertainties is possible. Key rules of transformation the information providing the automatic control and verification of data are formulated. As means of research and modeling of calculators it is offered to use Petri network.

Keywords: the software of Real-Time systems, model of calculations, workflow networks Petri, templates of object-oriented programming.

Туркін Ігорь Борисович – д-р техн. наук, проф., зав. кафедрой 603, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: energy@d4.khai.edu.

Соколова Евгения Витальевна – ст. преп. кафедры 603, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: sev_ai@mail.ru.