Интернет-журнал «Науковедение» ISSN 2223-5167 http://naukovedenie.ru/

Том 7, №3 (2015) http://naukovedenie.ru/index.php?p=vol7-3

URL статьи: http://naukovedenie.ru/PDF/124TVN315.pdf

DOI: 10.15862/124TVN315 (http://dx.doi.org/10.15862/124TVN315)

УДК 681.518

Веселов Алексей Аркадьевич

ФГБОУ ВПО «Тверской государственный технический университет» Россия, Тверь
Профессор кафедры «Электронные вычислительные машины»
Доктор технических наук
Доцент

E-mail: Veselov_A_A@mail.ru

РИНЦ: http://elibrary.ru/author_profile.asp?id=786013

Распределенная модель устройств цифровой и вычислительной техники на основе сетей Петри

_

¹ 170026, г. Тверь, наб. Афанасия Никитина, д. 22

Аннотация. Непрерывный рост сложности современных цифровых устройств и сокращение сроков их морального старения неизбежно приводит к тому, что разработчик уже не в состоянии гарантировать должного качества их функционирования. В таких условиях сильно возрастает значение моделей, позволяющих выявлять и устранять ошибки проектирования на самых разных стадиях разработки. Однако, по мере роста размеров модели, они становятся все более медленными. Причем настолько, что их использование становится либо затруднительным, либо уже практически бесполезным. В связи с этим остро встает вопрос о дальнейшем совершенствовании моделей.

Одним из перспективных путей для ослабления указанной проблемы является переход от монолитных (централизованных) моделей к их распределенным аналогам. В настоящее время уже имеется достаточно большой теоретический задел по построению распределенных систем, который можно эффективно использовать и для построения распределенных моделей.

В данной работе, на основе математического аппарата сетей Петри, дается формальное описание нового подхода к построению иерархической распределенной модели. Описывается новый взгляд на структуру компоненты, как состоящую из двух взаимосвязанных частей: функционального модуля и ее внешнего представителя.

Ключевые слова: дискретно-событийная система; сети Петри; интерфейс; удаленное взаимодействие; распределенная компонента; функциональный модуль; внешний представитель; входные и выходные полюса.

Ссылка для цитирования этой статьи:

Веселов А.А. Распределенная модель устройств цифровой и вычислительной техники на основе сетей Петри // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 7, №3 (2015) http://naukovedenie.ru/PDF/124TVN315.pdf (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/124TVN315

Объекты цифровой и вычислительной электронной техники (ЦВЭТ) можно отнести к одним из самых распространенных изделий, размеры и сложность которых непрерывно возрастают, а их номенклатура и алгоритмы функционирования постоянно изменяются и совершенствуются. Такое состояние дел приводит к необходимости постоянного совершенствования процессов их разработки и проектирования. Одним из основных инструментов повышения эффективности современных САПР являются модели. Они позволяют оценивать работоспособность и качество функционирования создаваемых устройств ЦВЭТ на самых разных стадиях проектирования и разработки еще до изготовления их физических макетов или опытных образцов. В результате, использование моделей позволяет сократить сроки и повысить качество проектирования новых или модифицируемых объектов.

В настоящее время для моделирования объектов ЦВЭТ наиболее широко используются так, называемые монолитные (централизованные) модели, К сожалению, они применимы для моделирования и анализа только относительно небольших и несложных систем или объектов. Это ограничение связано с влиянием, так называемой проблемы сложности, которая заключается в том, что рост количества состояний, в которых может оказаться моделируемый объект, увеличивается значительно более высокими темпами, чем его размеры. Кроме того, увеличение размеров моделей неизбежно приводит к снижению их быстродействия. При этом сами модели могут стать настолько медленными, что их практическое использование уже становится нецелесообразным. Перечисленные недостатки заметно ограничивают области применения монолитных моделей. В связи с этим, стали появляться многочисленные работы, направленные на исследование особенностей распределенных систем и разработку методов их построения и анализа [1-3]. Результаты анализа этих работ позволяют выделить два основных направления: системы распределенных вычислений и собственно распределенные системы.

Особенности поведения объектов ЦВЭТ позволяют отнести их к разряду дискретнособытийных систем (ДСС) для моделирования которых наибольшее применение получили математические аппараты теории сетей Петри (СП), теория автоматов и их разновидностей (графы состояний графы переходов и др.) [4,5]. Однако, среди всего этого многообразия, сети Петри выглядят наиболее перспективным аппаратом, имеющим интуитивно понятное графическое отображение и позволяющим более естественным образом моделировать явления параллелизма, синхронизации, гонки сигналов и т.д. К дополнительному достоинству СП следует отнести наличие у этого аппарата хорошо развитых аналитических возможностей. Имеется множество примеров эффективного применения СП для моделирования ЦВЭТ [5-7].

К распределенному моделированию, среди многочисленных модификаций СП появились такие как: СП высокого уровня [7], иерархические, вложенные, композиционные [9] и функциональные [10]. Характерной их особенностью являлась ориентации на использование модульного представления о распределенной компоненте в виде автономно функционирующего блока, обладающего внутренним состоянием и наличием входов и выходов. Наличие входов и выходов рассматривалось как чрезвычайно удобное средство для организации взаимодействия между компонентой и ее внешним окружением. Тем не менее, такой взгляд на компоненту как на основной элемент для построения распределенных систем, обладал одним очень заметным недостатком. Он заключался в том, что часть распределенной системы, представляемую в виде такой компоненты, можно было только непосредственно вложить в ее состав, не позволяя "оторвать" ее от места расположения самой системы и разместить где-нибудь в другом месте. Например, в другом приложении или на другом компьютере. Другими словами, место расположения распределенного модуля должно совпадать с местом его использования. По этой причине

описания многих СП, объявленных как распределенные, на самом деле были способны только визуально отобразить структуру модель как иерархическую. Однако, перед проведением имитационного эксперимента с моделью, ее многоуровневое описание приходилось снова преобразовывать в монолитную модель [11].

Появление и последующее развитие сетевых информационных технологий привело появлению еще одного направления исследований, направленных на поиск новых подходов к построению распределенных моделей с использованием СП, например [12]. Суть этих работ сводится к использованию механизмов удаленного взаимодействия при организации взаимодействия между базовыми элементами самой СП (позициями и переходами).

В данной работе предлагается новый взгляд на компоненту, который по сути дела комбинирует достоинства ранее существовавших подходов к пониманию сущности и назначения компоненты [13]. Принципиальное отличие этого подхода заключается в том, что распределенную компоненту следует рассматривать как две взаимосвязанные между собой части: блок функционального модуля (ФМ), реализующий необходимую функциональность (удаленный объект) и блок его внешнего представителя (ВП), как это было сделано в механизме "клиент-серверной" технологии удаленного взаимодействия. На полезность такого представления уже указывалось в некоторых ранних работах, например, в [14]. Предлагаемое разделение компоненты распределенной системы на две взаимосвязанные между собой части позволяет физически отделить ФМ компоненты от места его непосредственного использования. Взаимодействие же между функциональными модулями и их внешними представителями осуществляется только через их интерфейсы, представленные в виде соответствующих наборов входных и выходных позиций.

В качестве основы для построения распределенной модели была выбрана модификация СП, наиболее приспособленная для моделирования поведения ЦВЭТ и получившее название DPN [15]. Кроме всего прочего, DPN модель позволяет легко представлять ее в виде модуля с внутренними состояниями и набором входных и выходных полюсов. В общем виде данную разновидность СП можно представить как совокупность, состоящую из пяти множеств:

$$DPN=(P, T, F, D, M),$$
 (1)

где:

Р - конечное множество позиций;

T – конечное множество переходов ($P \cap T = \emptyset$);

 $F=T\times P \to link = \{ in, out, allow, forbid \}$ — конечное множество связей между позициями P и переходами T (in — входная дуга, out — выходная дуга; allow — разрешающая дуга и forbid — запрещающая дуга).

 $D = T \rightarrow \{ \tau \}$ — временные задержки на срабатывание переходов;

 $\mathbf{M} = \mathbf{P} \to \{\ \mathbf{0},\ \mathbf{1}\ \}$ — маркировка позиций, характеризующая конкретное состояние модели.

При этом на связи между узлами в DPN накладывается следующее ограничение:

$$\forall t \in T: |\operatorname{in}(t)| + |\operatorname{out}(t)| < 2, \tag{2}$$

где:

in(t) – множество входных позиций перехода t, $in(t) \subset P$;

 $\operatorname{out}(t)$ - множество выходных позиций перехода t, $\operatorname{out}(t) \subset P$ ($\operatorname{in}(t) \cap \operatorname{out}(t) = \emptyset$).

Более детально с этим расширением можно ознакомиться в [15].

Позиции, связанные с переходом разрешающей или запрещающей связью, названы соответственно разрешающей или запрещающей. Разрешающие и запрещающие связи между позициями и переходами получили название условных связей, а входные и выходные – активных связей. Если условные связи определяют условие срабатывания перехода, то активные связи определяют еще и результат его срабатывания путем изменения маркировки активной позиции. При наличии маркера в позиции, она раскрашивается темный цвет, а при его отсутствии – в белый (Рис. 1).

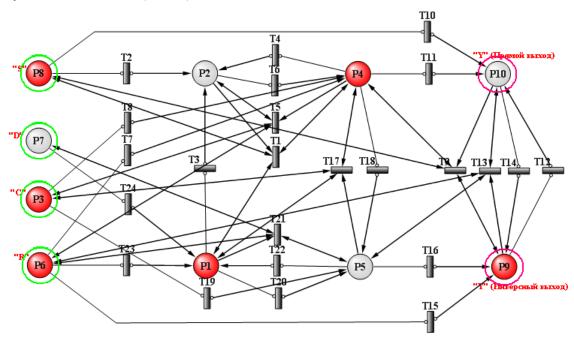


Рис. 1. DPN-модель триггера (микросхема К155ТМ2) (составлено автором)

Позиции, связанные с переходами DPN-модели с помощью только условных связей, являются ее входными позициями. Поскольку они определяют только условие, а не результат срабатывания возбужденных переходов, то срабатывание переходов, расположенных внутри модели, не способно изменить маркировку связанных с ними условных позиций. Это означает, что маркировку входных позиций невозможно изменить изнутри самой модели, а только со стороны ее внешнего окружения. Их назначение заключается только в том, чтобы передать модели воздействия со стороны ее внешнего окружения. В отличие от входных позиций, выходные позиции, назначаются в модели произвольно. Объясняется это тем, что выходные сигналы, моделируемые позициями модели, представляют собой только те внутренние состояния объекта, которые доступны для непосредственного наблюдения. При этом принимается, что множества входных и выходных позиций DPN модели не имеют общих элементов, т.е. никогда не пересекаются. На Рис. 1 входные и выходные позиции выделены круглым контуром.

Перечисленные особенности DPN-модели делают ее чрезвычайно удобной для формального описания функционального модуля, который можно представить в следующем виде:

$$\Phi M = (DPN, In, Out), \tag{3}$$

где:

DPN – D-сетевая модель, соответствующая определению (1);

In \subset DPN.P, Out \subset DPN.P − соответственно непересекающиеся множества водных и выходных позиций (In \cap Out = \varnothing).

Следует заметить, что позиции DPN-модели, не вошедшие в состав входных или выходных позиций, относятся к ее внутренним позициям, недоступным для их непосредственного наблюдения, а их маркировка отображает внутреннее состояние функционального модуля.

Внешний представитель такого ΦM , можно рассматривать как отдельный объект, состоящий из следующих множеств:

$$B\Pi = (Pin, Pout, Links)$$
 (4)

где:

Pin и **Pout** - непересекающиеся наборы входных и выходных полюсов внешнего представителя (**Pin** \cap **Pout** = \varnothing).

Links = (BП.Pin × Φ M.In) \cup (BП.Pout × Φ M.Out) — отношение связи между входными и выходными позициями Φ M и соответствующими полюсами его ВП.

Внешний вид ФМ и его ВП такой компоненты представлен на Рис.2. Визуально ВП ФМ изображается в виде прямоугольника с изображением полюсов, расположенных на его боковых сторонах. Причем слева отображаются входные полюса, а справа - выходные полюса. Наличие однозначного соответствия между полюсами ВП и входными и выходными позициями ФМ позволяет рассматривать ВП как его полноценный поведенческий аналог. Основная задача ВП сводится только к тому, чтобы отсылать сообщения об изменении маркировки своих входных полюсов соответствующим входным позициям ФМ и изменять маркировку своих выходных полюсов в ответ на сообщения об изменении маркировки выходных позиций ФМ.

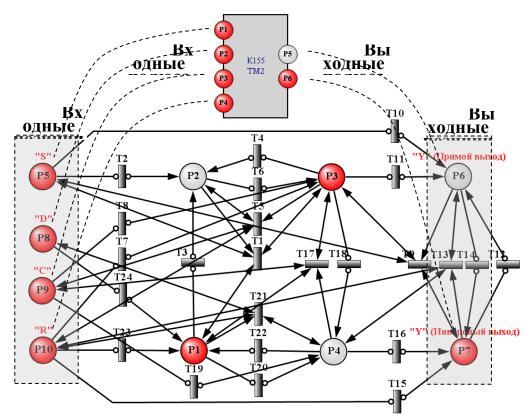


Рис. 2. DPN-модель функционального элемента триггера (К555TM2) и его внешнего представителя (составлено автором)

Формальное определение компоненты можно представить в виде объединения функционального модуля и его внешнего представителя:

$$\mathbf{K} = (\mathbf{\Phi}\mathbf{M}, \mathbf{B}\mathbf{\Pi}), \tag{5}$$

где:

ФМ - функциональный модуль компоненты;

ВП = (**Pin**, **Pout**) — внешний представитель функционального модуля, содержащий непересекающиеся наборы входных и выходных полюсов (**Pin** \cap **Pout** = \varnothing).

Становится очевидным, что представление компоненты в виде взаимосвязанных между собой частей имеет всё необходимое для того, чтобы использовать ее в составе другой, более сложной компоненты. Однако, для того, чтобы предоставить такую возможность, необходимо несколько изменить определение базовой DPN модели (1), добавив в ее состав множество ВП подчиненных компонент. В результате получим распределенный вариант DPN модели (DDPN), который можно представить в следующем виде:

$$DDPN = (DPN, B\Pi\Pi K), \tag{6}$$

где:

DPN - модель, соответствующая определению (1);

ВППК = $(\cup^{k}_{i=0} \ B\Pi i)$ — множество ВП удаленных ФМ подчиненных компонент, входные и выходные полюса которых входят в состав позиций владеющего ФМ $((In \cup Out) \subset P \ u \ In \cap Out = \emptyset)$.

При этом формальное определение распределенной версии функционального модуля, тоже изменится и примет несколько иной, чем (3) вид, а именно:

$$\mathbf{P}\mathbf{\Phi}\mathbf{M} = (\mathbf{D}\mathbf{D}\mathbf{P}\mathbf{N}, \mathbf{I}\mathbf{n}, \mathbf{O}\mathbf{u}\mathbf{t}),\tag{7}$$

где:

DDPN –распределенная модель, соответствующая определению (5);

In, **Out** - соответственно непересекающиеся множества входных и выходных позиций распределенной модели ((**In** \cup **Out**) \subset **DDPN.DPN.P** и **In** \cap **Out** = \varnothing).

При этом выражения (4) и (5), определяющее понятие ВП Φ М и самой компоненты, остаются без изменений.

Дальнейшее рассмотрение распределенной (DDPN) модели будем проводить на конкретном примере. На Рис. 3 показано условное изображение функционального элемента триггера (К555ТМ2) и структурная схема его реализации. Структурная схема включает в свой состав шесть функциональных логических элементов "ЗИ-НЕ", между которыми установлены соответствующие соединения, обеспечивают реализацию которые, И (функциональность) При проектировании триггера В целом. схемного решения проектировщик использует только условное изображение функционального элемента. Но при необходимости он всегда может перейти к его более детальному представлению в виде соответствующей структурной схемы. Это указывает на то, что предложенный подход, связанный с представлением распределенной компоненты в виде двух семантически связанных между собой частей, уже давно был известен и эффективно использовался в области проектирования схемных решений. В данном случае его новизна заключается только в переносе ранее известного подхода из области схемных решений в новую предметную область, связанную с системами распределенного имитационного моделирования.

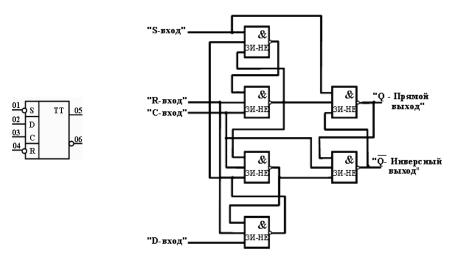


Рис. 3. Условное изображение триггера (а) и структурная схема его реализации (б) (составлено автором)

Распределенная DDPN модель этого триггера показана на Рис. 4. Для удобства дальнейших рассуждений, ФМ, содержащий в своем составе ВП ФМ подчиненных компонент, будем называть владеющим ФМ. А ФМ, внешние представители которых, входят в состав других ФМ, будем называть подчиненными. Как видно из Рис. 4, DDPN-модель состоит из владеющего ФМ (в центре рисунка), реализующего логику триггера, и шести подчиненных ей компонент, реализующих поведение логических элементов "ЗИ-НЕ". Взаимодействие ФМ модели триггера со своим внешним окружением осуществляется через его внешнего представителя, показанного в верхней части рисунка. Совокупность ФМ DDPNмодели триггера и его ВП представляет собой распределенную компоненту. Как легко заметить, на верхнем иерархическом уровне распределенная DDPN-модель триггера представляется в виде ФМ, в состав которого входят шесть ВП подчиненных ей компонент $(B\Pi_1-B\Pi_6)$. Кроме них в состав владеющего ΦM модели триггера входят переходы и дуги, соединяющие их с полюсами ВП ФМ подчиненных компонент. По сути дела содержимое ФМ, кроме ВП ФМ подчиненных компонент, представляет собой обычную DPN-модель (т.е. модель). основное назначение которой заключается в обеспечении коммуникации между ВП подчиненных компонент.

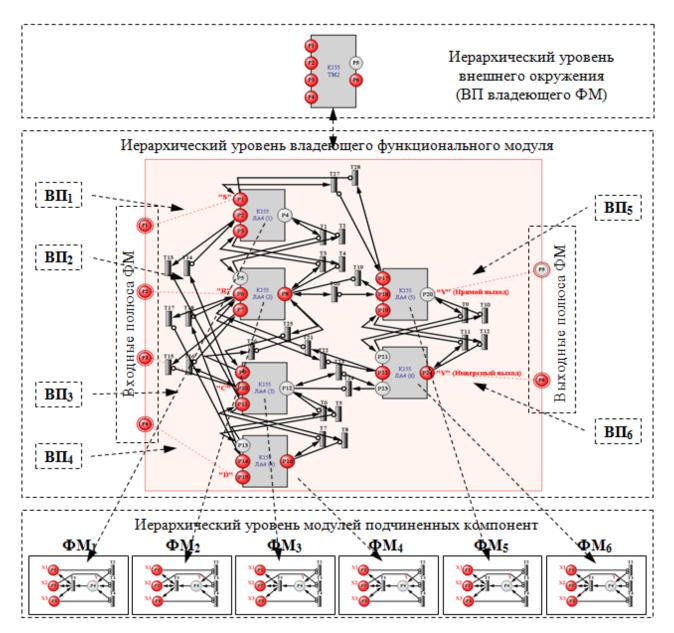


Рис. 4. Структура распределенной модели триггера KT555TM2 (составлено автором)

Поскольку ВП подчиненной компоненты является полным поведенческим аналогом представляемого им ФМ, то владеющий ФМ ведет себя с ними точно так же, как он вел бы себя с представляемыми ими ФМ. Основное назначение ВП подчиненных ФМ заключается в передаче своему ФМ информации об изменении маркировки его входных полюсов. В свою очередь задача ФМ подчиненных компонент заключается в управлении своим поведением и передаче информации об изменении маркировки своих выходных позиций внешнему представителю. Это позволяет рассматривать ФМ распределенной компоненты как блок, реализующий функциональные возможности компоненты, а его ВП — как блок, предоставляющий функциональные возможности этого ФМ для их непосредственного использования внешним окружением.

Если в составе DDPN-модели отсутствуют подчиненные компоненты, то она превращается в обычную монолитную DPN-модель. Отсутствие у Φ M подчиненных компонент указывает на то, что она находится на самом нижнем иерархическом уровне. В распределенной модели триггера (Рис. 4) такими функциональными модулями являются модули Φ M₁- Φ M₆. Очевидно, что для включения компоненты в состав другого более

сложного функционального модуля, достаточно ввести внешний представитель подчиненной компоненты в состав этой владеющей компоненты.

Алгоритм функционирования распределенной модели разбивается на две фазы: инициализация и выполнение.

Этап инициализации необходим для приведения модели в исходное рабочее состояние и состоит из последовательного выполнения следующих шагов:

- 1. Просматриваются все переходы ФМ владеющей компоненты распределенной модели с целью поиска возбужденных переходов и определения времени, оставшегося до их срабатывания.
- 2. Определение минимального интервала времени (ΔТмин), оставшегося до срабатывания возбужденных переходов в коммуникационной DPN модели владеющего ΦМ.
- 3. Если у ФМ нет подчиненных компонент, то переходим к пп.8.
- 4. Произвольным образом выбирается ВП такой подчиненной компоненты, которая еще не находится в рабочем режиме. Выбранному ВП дается указание на передачу своему ΦM сообщения, интерпретируемое им как команда на выполнение с указанием временного шага, равного нулю (dt=0).
- 5. В ответ на сообщение о запуске, ФМ подчиненной компоненты выполняет все действия, начиная с пп.1 и возвращает значение минимального интервала времени, оставшееся до срабатывания его возбужденных переходов.
- 6. Определение минимального значения времени, оставшегося до срабатывания возбужденных переходов в коммуникационной DPN модели ФМ владеющей компоненты и в ФМ проверенных подчиненных компонент.
- 7. Если не все модели подчиненные компонент переведены в рабочий режим, то переходим к пп.4.
- 8. Определение минимального значения интервала времени, оставшегося до срабатывания возбужденных переходов в коммуникационной DDPN модели и ФМ подчиненных компонент.
- 9. Модель ФМ владеющей компоненты переводится в рабочий режим.

В результате такой инициализации распределенной модели она располагает безопасной величиной интервала, на который можно продвинуть модельное время в каждой из ее компонент. При этом сама модель приводится в состояние готовности к функционированию в своем рабочем режиме, алгоритм которого состоит из выполнения следующих шагов:

- 1. Распределенной модели подается команда на выполнение с передачей информации о шаге продвижения модельного времени ∆Тмин.
- 2. Устанавливаем ожидаемое значение времени, оставшееся до срабатывания возбужденных переходов в ФМ компоненты, равным нулю (∆Тмин.ож.= -1). Это означает, что в распределенной модели нет ни одного возбужденного перехода, а она сама находится в устойчивом состоянии.
- 3. Если в коммуникационной модели (ФМ компоненты) нет ни одного возбужденного перехода, то переходим к пп.4.
- 4. Если среди возбужденных переходов коммуникационной модели ФМ есть такие, у которых время, оставшееся до их срабатывания равно заданному

∆Тмин, то они срабатывают и выводятся из состава возбужденных переходов. (С правилами срабатывания переходов в DPN модели можно ознакомиться в [15] и здесь не рассматриваются). При этом, время, оставшееся до срабатывания остальных переходов, уменьшается на величину заданного шага изменения модельного времени.

- 5. Если в результате выполнения срабатывающих переходов в коммуникационной модели происходит изменение маркировки выходных позиций ФМ, то он генерирует сообщение своему ВП о необходимости изменения маркировки его выходных полюсов. Если же изменяется маркировка внутренних позиций, входящих в состав входных полюсов ВП подчиненных компонент, то они посылают соответствующее сообщение своему удаленному ФМ о необходимости изменения маркировки его входных позиций.
- 6. Если ФМ компоненты не располагает подчиненными компонентами, то переходим к пп.9.
- 8. В ответ на это сообщение ФМ подчиненной компоненты выполняет все действия, начиная с пп.2 и возвращает величину ожидаемого минимального значения временного интервала (ΔТмин.ож.комп.), оставшегося до срабатывания возбужденных переходов в его модели после ее выполнения с указанным шагом (ΔТмин).
- 9. Если в результате выполнения ФМ подчиненной компоненты он возвращает значение интервала, оставшегося до срабатывания ее возбужденных переходов (ΔТмин.ож.комп.) меньшее, чем ΔТмин.ож, то корректируем его следующим образом: ΔТмин.ож.= ΔТмин.ож.комп.
- 10. Если ∆Тмин.ож. больше или равен нулю, то ∆Тмин=∆Тмин.ож. Переходим к пп.1.
- 11. Остановка выполнения модели, поскольку в нет ни одного возбужденного перехода и поэтому все внутренние динамические процессы в ней завершены и она находится в устойчивом состоянии.

В дальнейшем при любой попытке изменить маркировку любой из позиций главного функционального модуля распределенной модели приведет к формированию новой команды на выполнение модели. При этом ее работа возобновляется, начиная с пп.1.

Таким образом, в данной работе представлена новая концепция построения распределенной модели и дано ее формальное описание на основе сетей Петри. Основное ее отличие от существующих подходов заключается в представлении компоненты в виде двух взаимодействующих между собой частей: функционального модуля и его внешнего представителя. Функциональная часть компоненты представляет собой блок, реализующий (функциональность) компоненты, а представительская часть блок, предоставляющий возможность использования внешним окружением этой функциональности.

Такое представление о компоненте позволяет:

- физически отделить ее функциональную и представительскую части друг от друга и размещать их не только в одном приложении, но и в разных приложениях и даже на разных компьютерах;
- создавать иерархические модели и расширять их как по горизонтали (путем вставки функционального модуля новой компоненты или путем замещения внешнего представителя подчиненной компоненты ее функциональным модулем), так и по вертикали;
- получать из распределенной модели ее монолитный аналог путем замещения всех внешних представителей подчиненных компонент их соответствующими удаленными функциональными модулями;
- осуществлять одновременную разработку дискретно-событийных систем по частям, независимо друг от друга;
- проектировать распределенные модели как сверху вниз, так и снизу верх;
- исследовать поведение большой и сложной системы на разных уровнях детализации;
- повысить быстродействие имитационных моделей сложных дискретнособытийных систем.

ЛИТЕРАТУРА

- 1. Э. Танненбаум, М. Ван Стеен. Распределенные системы. Принципы и парадигмы. СПб: Питер 2003 г. 877 с.
- 2. Черемисинов Д.И. Морфизмы моделей поведения распределенных систем. Информатика, январь-март, №1. 2005. стр. 76-88.
- 3. Александров А.А. Распределенное имитационное моделирование: технологии, методы, средства. Вестник НГУ. Серия: Информационные технологии. 2009. Том 7, Выпуск 3. стр. 62-69.
- 4. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. СПб.: Наука, 2000. 694 с.
- 5. Котов В.Е. Сети Петри. М. Наука Главная редакция физико-математической литературы, 1984. 160 с.
- 6. Варшавский В.И. и др. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах. Под ред. Варшавского В.И. М. Наука, 1986. 400 с.
- 7. Cortadella J., Rishinevsky V., Lavagno L., Yakovlev A. Logic synthesis of asynchronous controllers and interfaces. Springer (2002). 273 p.
- 8. Jensen K. Colored Petri Nets Basic Concepts, Analysis Methods and Practical Use. Springer-Verlag, Vol. 1-3, 1997.
- 9. Jiacun Wang. Timed Petri Nets. Theory and Application.// Kluwer Academic Publishers. 1998.
- 10. Zaitsev D.A. Functional Petri Nets. Universite Paris-Dauphine, Cahier du Lamsade 224, April 2005.
- 11. Непомнящий В.А., Попова Н.С. Моделирование спецификаций распределенных систем на языке Dynamic-Real сетями Петри высокого уровня. Вестник НГУ. Серия: Информационные технологии. 2010. Том 8, выпуск 4 (ISSN 1818-7900).
- 12. Дубинин В.Н. Распределенная реализация интерпретированных сетей Петри в архитектуре IEC 61499 // Межвуз. сб. науч. трудов "Вычислительные системы и технологии обработки информации", выпуск 3(29). Изд-во Пензенского гос. ун-та, 2005. с. 58-64.
- 13. Веселов А.А. Компонентный подход к построению распределенных имитационных моделей. "Программные продукты и системы". Международное научно-практическое приложение к международному журналу "Проблемы теории и практики управления". №2(98), 2012. с. 16-18.
- 14. Gomes L., Barros Joao-Paulo. Using hierarchical mechanisms with Petri nets for PLD based system design. The International Workshop on Discrete-Event System Design, DESDes'01, June 27÷29, 2001; Przytok near Zielona Gora, Poland.
- 15. Веселов А.А. Моделирование устройств цифровой электронной техники D-расширением сетей Петри. Монография. Тверь: ТГТУ, 2006.

Рецензент: Юдицкий Семен Абрамович, доктор технических наук, гл. научный сотрудник института проблем управления (ИПУ РАН) им. В.А. Трапезникова.

Veselov Aleksey Arkad'evich

State Educational institution of Higher Professional Education «The Tver state university»
Russia, Tver
E-mail: Veselov_A_A@mail.ru

Distributed model of digital and computing technical units on Petri nets base

Abstract. Continuous growth in the complexity of modern digital devices and shortening their obsolescence inevitably leads to the fact that the developer is not able to guarantee proper quality of their functioning. In such circumstances, greatly increased the value of models to identify and eliminate design errors at various stages of development. In addition, the growth of the size of the model they are getting slower. So much so that their use is either difficult or is almost useless. In this regard, sharply raises the question of further improving the models.

One of the promising ways to mitigate this problem is to move away from monolithic (centralized) models to their distributed peers. Currently, there is already a big enough theoretical groundwork for building distributed systems that can be effectively used for building distributed models.

In this paper, based on the mathematical apparatus of Petri nets, formal description is given of a new approach to the construction of a hierarchical distributed model. We describe a new look at the structure of the components, as consisting of two interconnected parts: a functional module and its external representative.

Keywords: discrete event system; Petri nets; interface; remote interoperation; distributed component; functional module; proxy.

REFERENCES

- 1. Tanenbaum A.S., Maarten van Steen. Distributed systems. Principles and paradigms. SPb: Piter. 2003. 877 p.
- 2. Cheremisinov D.I. Morphizms of behavior models of distributed systems. Informatic, January-march, №1. 2005. pp. 76-88.
- 3. Alexandrov A.A. Distributed simulation modeling: technology, methods, instruments. Vestnic NGU. Seria: Information technology. 2009. Vol. 7. Item 3, pp. 62-69.
- 4. Shalito A.A. Logical control. Methods of apparatus and program realization of algorithms. SPb.: Nauka, 2000. 694 p.
- 5. Kotov V.E. Petri nets. M.: Nauka. Glavnaia redakcia of physically-mathematical literature, 1984. 160 p.
- 6. Varshavski V.I. and other. Automatic control of asinchronabled processes in computers and discrete systems. M.: Nauka, 1986. 400 p.
- 7. Cortadella J., Rishinevsky V., Lavagno L., Yakovlev A. Logic synthesis of asynchronous controllers and interfaces. Springer (2002). 273 p.
- 8. Jensen K. Colored Petri Nets Basic Concepts, Analysis Methods and Practical Use. Springer-Verlag, Vol. 1-3, 1997.
- 9. Jiacun Wang. Timed Petri Nets. Theory and Application.// Kluwer Academic Publishers. 1998.
- 10. Zaitsev D.A. Functional Petri Nets. Universite Paris-Dauphine, Cahier du Lamsade 224, April 2005.
- 11. Nepomniachie V.A., Popova N.S. on Dynamic-Real language with help high level Petri nets. 2010. Vol.8, (ISSN 1818-7900).
- 12. Dubinin V.A. Distributed realization of interpreted Petri nets in IEC 61499 architecture. Interinstitute sbornic of science works "Computing systems and technology of information's processing", 3(29). Penza state university, 2005. p. 58-64.
- 13. Veselov A.A. Component approach to building the distributed simulate models. "Program products and systems" [Addition to Inernational magazine "Problems of theory and practice of control"]. №2(98), 2012. pp. 16-18.
- 14. Gomes L., Barros Joao-Paulo. Using hierarchical mechanisms with Petri nets for PLD based system design. The International Workshop on Discrete-Event System Design, DESDes'01, June 27÷29, 2001; Przytok near Zielona Gora, Poland.
- 15. Veselov A.A. Modeling digital electronic devices by D-extension of Petri nets. Monograph. Tver^ TSTU, 2006. 104 p.