

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 519.876.5

А.С.Сорокин

ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ И АЛГЕБРА ПРОЦЕССА МОДЕЛИРОВАНИЯ ХАРАКТЕРИСТИК

1. Введение

В работе рассматривается область характеристики моделирования, представлены стандартные парадигмы для определения стохастической характеристики модели, сети организации очередей и стохастические сети Петри.

Под парадигмой программирования будем понимать совокупность идей и понятий, определяющих стиль написания программ. Парадигма, в первую очередь, определяется базовой программной единицей и самим принципом достижения модульности программы.

В качестве этой единицы обычно выступают: определение (декларативное, функциональное программирование); действие (императивное программирование); правило (продукционное программирование); диаграмма переходов (автоматное программирование) и другие сущности. В современной индустрии программирования очень часто парадигма программирования определяется набором инструментов программиста, а именно, языком программирования и используемыми библиотеками.

Термин «парадигма программирования» впервые применил Роберт Флойд [1] в своей лекции лауреата премии Тьюринга. Флойд отмечает, что в программировании можно наблюдать явление, подобное парадигмам Кунна, но, в отличие от них, парадигмы программирования не являются взаимоисключающими: Если прогресс искусства программирования в целом требует постоянного изобретения и усовершенствования парадигм, то совершенствование искусства отдельного программиста требует, чтобы он расширял свой репертуар парадигм.

По мнению Роберта Флойда, в отличие от парадигм в научном мире, описанных Куном, парадигмы программирования могут сочетаться, обобщая инструментарий программиста.

Существуют различные определения «парадигмы программирования». Далеко не все авторы, использующие термин «парадигма программирования», решаются дать интенсиональное определение данному термину. Однако и те определения, которые удается найти, серьезно отличаются друг от друга.

Диомидис Спинеллис [2] даёт следующее оп-

ределение: Слово «парадигма» используется в программировании для определения семейства обозначений (нотаций), разделяющих общий способ (методику) реализаций программ. Для сравнения тот же автор приводит определения из других работ.

В статье Дэниела Боброва [3] парадигма определяется как «стиль программирования как описания намерений программиста». Брюс Шрайвер [4] определяет парадигму программирования как «модель или подход к решению проблемы», Линда Фридман [5]—как «подход к решению проблем программирования». Памела Зейв [6] даёт определение парадигмы как «способа размышлений о компьютерных системах».

Питер Вегнер [7] предлагает другой подход к определению термина парадигмы программирования. В его работе парадигмы определяются как «правила классификации языков программирования в соответствии с некоторыми условиями, которые могут быть проверены».

Тимоти Бадд [8] предлагает понимать термин «парадигма» как «способ концептуализации того, что значит „производить вычисления“, и как задачи, подлежащие решению на компьютере, должны быть структурированы и организованы».

В разделе 3 представлена алгебра процесса и в последующих подразделах рассмотрены некоторые из расширений вероятностных процессов. В частности, описаны Исчисления Коммуникационных Систем (CCS) и основанные на них различные расширения. В [9-13] обсуждена связь алгебры процесса с оценкой характеристик.

2. Моделирование характеристик

Для описания анализа и оптимизации динамического поведения компьютера и систем коммуникации необходимо оценить характеристики. Это потребует исследования потока данных и информации контроля как внутри, так и между компонентами системы. Цель состоит в том, чтобы понять поведение системы и идентифицировать аспекты системы, которые с точки зрения характеристик наиболее чувствительны.

Моделируя абстрактное представление или модель системы, в характеристиках отражают

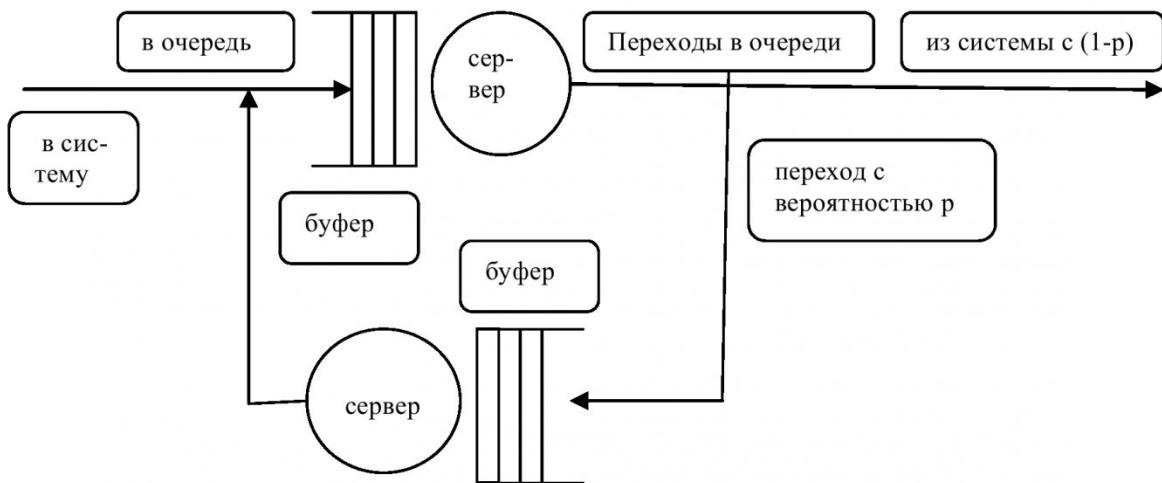


Рис. 1: Простая открытая сеть организации очередей

существенные особенности системы так, чтобы они могли быть воспроизведены. Это позволяет представить несколько альтернатив для значений, данных параметрам модели.

Определить поведение модели и критерий качества работы возможно по текущим множествам значений параметров. Оценка может быть произведена с помощью решения системы уравнений, которое реализовано аналитическим или численным методом, или с помощью моделирования. Аналитические модели обычно основаны на стохастических моделях и в последующем понятие моделирование характеристик не относится к стохастическим моделям, разрешенным аналитически или установленным иначе.

Существуют две системы обозначений для того чтобы построить такие модели - сети организации очередей и стохастические сети Петри. Они описаны соответственно в разделах 1. и 2. Во многих случаях эти основные стохастические модели являются марковскими процессами.

Размерность и сложность многих современных систем приводят к большим сложным моделям. Это привело к композиционному подходу моделирования характеристик. При этом подходе систему преобразуют в подсистемы, которые легче смоделировать. Ряд авторов приняли программное обеспечение, проектируя стиль, структурирующий методы для построения модели характеристик [14].

Трудность в исследовании анализа характеристик [15] составило обнаружение методов для решения больших цепей Маркова, пространства состояний которых конечны, но чрезвычайно велики. Так как стандартные численные методы не могут справиться с такими моделями, то в задачах часто появляется состояние всплеска в промежутке. Отыскиваются композиционные подходы, которые были бы применимы к модельному решению так же, как и типовая конструкция, позволяющая отыскивать раздельные реше-

ния подмоделей. В целях решения модель нужно рассматривать как единое целое, но в этом пункте предложен метод, который позволяет подсистемам быть смоделированными раздельно.

Представлены некоторые подходы к модельному упрощению, которые могут быть применены отдельно к подсистемам модели, но которые не затрагивают целостности всей модели.

2.1. Сети. Организации очередей

Известно использование сетей организации очередей для моделирования характеристик. В этом разделе кратко представлены основные идеи и некоторая терминология, которая будет применена позднее. Более подробно это изложено в [16].

Очередь состоит из процесса прибытия, буфера, где клиенты ждут обслуживания, и одного или более серверов, представляющих ресурс, который должен быть сохранен для каждого клиента в течение некоторого периода прежде, чем оставить очередь.

Очередь может быть охарактеризована пятью факторами: оценкой прибытия, оценкой обслуживания, числом серверов, ёмкостью буфера и дисциплиной организации очередей (рис.1).

Первые четыре из этих факторов могут быть кратко представлены *системой обозначений Кендала*, для того чтобы классифицировать очереди. В этой системе обозначений очередь представлена как $A/S/c/m/N$. Процесс прибытия обычно обозначают M , что означает марковский процесс (показательную функцию), G общие или D детерминированные распределения. Могут также использоваться идентификаторы для других распределений, таких как H_k (гиперэкспоненциальное распределение с параметром k). S - оценка обслуживания, использующая идентификаторы распределения. c - число серверов, дос-

тупных для оказания услуг очереди. m - ёмкость буфера по умолчанию. Когда буфер полон, прибывающие клиенты могут быть потеряны или заблокированы. По умолчанию N - конечная численность клиентов.

В случае по умолчанию последние два классификатора могут быть опущены. Порядок организации очередей определяется тем, как сервер выбирает клиента из очереди для последующего обслуживания. Например, порядок может быть таков: *начало-прибытие-начало-подача* (FCFS), в котором у клиента было самое длинное ожидание, или *разделяющий процессор* (PS), который способен обслуживать раздельно всех клиентов, поставленных в очередь.

Сеть организации очередей это ориентированный граф, в котором узлы - очереди, называемые *центрами обслуживания*, представляют ресурс в смоделированной системе (Рис.2). Клиенты, представляющие рабочие места в системе, проходят через систему и конкурируют с её ресурсами. Дуги сети представляют топологию системы, и вместе с *направлением вероятностей* определяют пути, по которым клиенты проходят через сеть. В зависимости от требований, предъявляемым к ресурсам и оценкам обслуживания, может возникнуть потребность в ресурсе, которая приведет к формированию очереди ждущих клиентов.

Состояние системы представляется числом клиентов, занятых в каждом из центров обслуживания. Существуют различные *классы* клиентов, имеющих различные особенности в сети. В этом случае знаем численность клиентов каждого класса в каждом центре обслуживания. Сеть может быть *закрытой*, *открытой* или *смешанной* в зависимости от того, остается ли фиксированной численность клиентов в системе. Клиенты могут или прибыть от, или отбыть к внешней окружающей среде; или найдутся классы клиен-

тим, имеющих прямое в вычислительном отношении эффективное решение [17]. Этот класс исключает некоторые важные особенности системы, которые позволяют получить критерии качества работы без обращения к основному марковскому процессу. Решения этих моделей, которые часто называют решением в виде *произведения*, позволяют рассматривать раздельно индивидуальные очереди в сети.

2.2. Обобщенные стохастические сети Петри

Сети Петри представляют собой направленные графы с двумя типами узлов, *мест* и *переходов*, и односторонних дуг между ними. Символы перемещаются между местами согласно указанным направлениям движения при переходах. Переход возникает, когда каждое из мест, связанных с ним, имеет не менее одного символа; если это так, то переход удаляет символ из каждого из этих мест и вносит символ в каждое из мест, с которым оно связано.

Состояние системы определяется числом символов в каждом месте сети. Это называют *маркировкой* сети. Сеть Петри определяется ее структурой и начальной буквой *маркировки*, которая является начальным размещением символов. *Множество достижимости* - это множество всех возможных маркировок, которые имеются в сети, после отмены правила начинаяющихся с начальной маркировки. Это используется естественным образом для формирования *графа достижимости*.

Различные вычислительные и стохастические расширения сетей Петри были предложены для характеристики моделирования [18]. Среди наиболее распространенных были стохастические сети Петри (SPNs), предложенные в [19] и их последующее развитие - обобщенные стохастические сети Петри (GSPNs) [20].

В SPNs оценка включения (возможно зави-

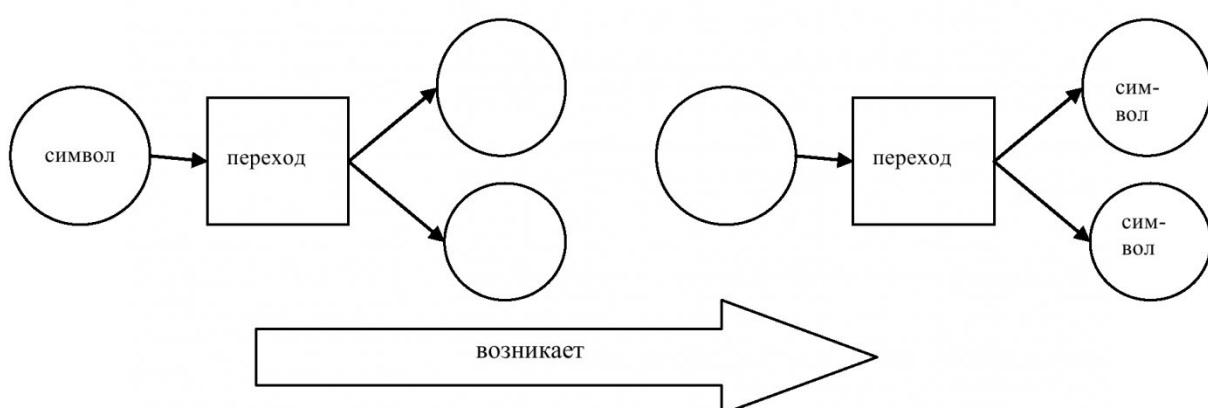


Рис. 2. Простое включение Сети Петри

тов, показывающие соответственно открытые и закрытые шаблоны поведения.

Большой класс организации очередей в сетях

связан с маркировкой) распределена по экспоненте и связана с каждым переходом. Как только переход разрешен (отмечено каждое входное

место), по графику распределения можно определить задержку прежде, чем реализуется переход; если же переход все еще разрешен в конце этого времени, то тогда он будет осуществляться.

Было показано, что граф достижимости, лежащий в основе таких сетей, изоморфен марковскому процессу, когда эта задержка распределена по экспоненте [21].

Таким образом, SPNs обеспечивает альтернативное средство определения стохастической модели, используемой для моделирования характеристик. Кроме того, SPNs в состоянии легко выразить некоторые особенности, такие как использование множественного ресурса, которые трудно смоделировать в сетях организации очередей. Критерии качества работы обычно получаются из моделей с помощью численного решения основного марковского процесса.

Были получены решения в виде произведения для SPNs, например [22], но они используют ограничительные условия на структуру сети.

В GSPNs переходы сети разделены в два подмножества - *рассчитанные* переходы, которые ведут себя как переходы в SPNs, с распределенным по экспоненте временем запуска, и *непосредственные* переходы, которые позволяют запуск. Разрешен непосредственный запуск переходов перед любыми рассчитанными переходами. Следовательно, граф достижимости GSPN может быть разделен на *материальные и обращающиеся в нуль маркировки*. Показано, что, так как никакое время не протекает в обращающихся в нуль маркировках, то они могут быть устраниены еще до решения вложенной цепи Маркова. Итак, непосредственные переходы игнорируются во время модельного решения.

Модели GSPN широко использовались для анализа характеристик, например [23]. Так же как непосредственные переходы GSPNs иногда включают дуги замедлителя. Такие расширения в системе обозначений часто позволяют выразить модель более кратко, но они не увеличили мощность моделирования GSPNs [24].

Стochasticные сетевые графики (SAN), представленные в [25], интересны тем, что, как и PEPA, они подчеркивают действия системы. Подобно GSPNs эти сети, предназначенные для моделирования характеристики (совместное рассмотрение характеристик и пригодности системы), имеют большие структуры. Так как медленный переход и замедлитель образуют дугу, они включают в сеть *схемы и случаи*, которые представляют более сложный запуск правил. В [26] представлена абстрактная модель, *стохастическая система деятельности*, которая используется для применения SAN. В [14] исследованы композиционные методы для SAN.

3. Алгебра процесса

Алгебра процесса это математическая теория,

которая предоставляет математический аппарат для модельных параллельных систем, для того чтобы исследовать структуру и поведение модели. Примерами являются Исчисление Коммуникационных Систем (CCS) [27], Коммуникационные Последовательные Процессы (CSP) [28] и Алгебра Коммуникационных Процессов (ACP) [29]. Система характеризуется ее активными компонентами и *взаимодействиями*, или *коммуникациями* между ними. В отличие от сетей организации очередей или сетей Петри в них нет понятия *объекта* или *потока* в модели. Композиционное доказательство является неотъемлемой частью языка программирования.

В CCS активные компоненты системы называют *агентами* или *процессами*, и они предпринимают *действия*, представляющие дискретные действия системы. Любое действие может быть внутренним агентом или может составить взаимодействие или коммуникацию между соседними агентами. Агенты могут возобновить свои внутренние действия одновременно, но важно отметить, что это поведение определяет семантику чередования.

Комбинаторы языка дают возможность построения фактора, у которого есть определяемое первое действие (префикс); имеется альтернативный выбор (выбор); или имеются параллельные возможности (композиция). В языке PEPA сохраняются префикс и выбор, а композиция заменена на *кооперацию*.

Как и многие другие алгебры процесса, CCS даёт рабочую семантику, в стиле [30], используя маркированную систему перехода. Может быть построено *производное дерево*, или *граф*, в котором языковые элементы формируют узлы, а переходы - дуги. Эта структура является инструментом, для того чтобы исследовать факторы и системы, которые они представляют. Это есть основа стиля *бимоделирования эквивалентности*. Таким образом, два агента эквивалентны, если они выполняют точно те же самые действия. Сильные и слабые формы эквивалентности определяются в зависимости от внутренних действий агента. Бимоделирование и связанные с ним понятия эквивалентности представлены более подробно в [9-13].

Модели CCS использовались для установления поведения обеих систем относительно данной спецификации, а также и в более абстрактном смысле. Это называют *операторным* или *качественным* моделированием. Исследованы поведенческие свойства, такие как *равнодоступность* и *свобода от взаимной блокировки*, в отличие от количественных значений, полученных из характеристик модели.

В следующих разделах будут обсуждены некоторые расширения, сделанные для того чтобы включить времена и вероятности в алгебры процесса. Большинство из них

может быть показано расширением CCS. При обращении к алгебре процесса без таких расширений называют это *чистой* алгеброй процесса.

3.1. Временное расширение алгебры процесса

В чистой алгебре процесса предполагается, что все действия были мгновенными, и только относительный выбор времени представлен с помощью проекции прямого процесса.

Самый простой путь, по которому время может быть включено в такую алгебру, это создание синхронизации.

В синхронных исчислениях, таких как SCCS [31], принятые неявные глобальные часы, и одно действие должно происходить с каждым тиком часов. Однако чтобы смоделировать оперативное поведение систем, необходимо более сложное представление о времени.

Как было указано во временной CCS (TCCS) [32] время, разрешающее агенту зафиксировать периоды задержки указанных длин в дополнение к фиксированным действиям, может быть представлено явно в алгебре процесса.

Предполагается, что действия TCCS мгновенны, и времена исчисляются натуральными числами. Языку придаётся рабочая семантика с двумя различными типами переходов: переходы действия и переходы времени.

В режиме реального времени ACP [33] проявляется альтернативный подход. *Абсолютное* время связано с каждым случаем, где случай является завершением действия в процессе.

Можно определить относительное время каждого действия, или интервал, во время которого случай должен произойти. Такие интервалы позволяют ввести оператор интегрирования, так как он предоставляет выбор континуума альтернатив.

3.2. Вероятностные алгебры процесса

Алгебры процесса часто используют в модельных системах, в которых есть неопределенность в поведении компонент, но со временем абстрагируются от этой неопределенности так, что все выборы становятся недетерминированными.

Вероятностные расширения алгебр процесса позволяют эту неопределенность определить ко-

личественно, так как недетерминированный выбор заменяется вероятностным. В этом случае вероятность ассоциирована с каждым возможным результатом выбора.

Рабочая семантика для вероятностных алгебр процесса даётся в терминах вероятностных маркированных систем перехода.

В маркированных системах перехода вероятности ассоциируются с переходами. Эти системы классифицируются как *реактивные* или *порождающие*.

В реактивной системе вероятности переходов агента могут зависеть от окружающей среды, в которой помещен агент.

В порождающей системе вероятности перехода независимы от окружающей среды. В действительности, в реактивном случае распределение вероятности определено по возможным производным агента, выполнившего специфическое действие, а в порождающем случае распределение вероятности определено по возможным действиям агента.

В [34] описан язык PCCS, который подобен SCCS, но с вероятностным выбором, заменяющим недетерминированный выбор. Другое расширение SCCS - это разделы WSCCS [35], в которых используются веса для определения вероятностей. Здесь недетерминированный выбор заменен на вероятностный и расставленный по приоритетам выбор.

Вероятностные алгебры процесса были предложены как наиболее подходящий способ проверки эквивалентности между спецификой системы и ее выполнением [36]. Два процесса по всей вероятности будут бигомотетичными или эквивалентными, если их видимое поведение будет тем же самым с вероятностью $1-\varepsilon$, где ε бесконечно малая величина. Другая альтернатива – использование предположений, выражающих идею, что один процесс может быть по всей вероятности лучше, чем другой [37]. Это необходимо, для того чтобы показать, что выполнение системы улучшается. Таким образом, если требуется вероятность невыполнения 0.05, то выполнение, которое гарантирует, что вероятность невыполнения меньше, чем 0.04, будет удовлетворительной.

СПИСОК ЛИТЕРАТУРЫ

1. Floyd R. W. *The Paradigms of Programming*. *Communications of the ACM*, 22(8):455—460, 1979. Русский перевод см. в кн.: Лекции лауреатов премии Тьюринга за первые двадцать лет (1966—1985), М.: МИР, 1993.
2. Spinellis D. D. Programming paradigms as object classes: a structuring mechanism for multiparadigm programming. *PhD thesis*, University of London, London SW7 2BZ, United Kingdom, February 1994.
3. Bobrow D. G. If Prolog is the answer, what is the question. // *Fifth Generation of Computer Systems*, pages 138—145, Tokyo, Japan, November 1984. Institute for New Generation Computer Technology, North-Holland.
4. Shriver B. D. Software paradigms. *IEEE Software*, 3(1):2, January 1986.
5. Friedman L. W. Comparative programming languages: generalizing the programming function. 1991.
6. Zave P. A compositional approach to multiparadigm programming. *IEEE Software*, 6(5): 15—25, 1989.

7. Wegner P. Concepts and paradigms of object-oriented programming. *{OOPS} messenger*, 1(1): 7—87, 1990.
8. Budd T. A. Multy-Paradigm Programming in LEDA. Addison-Wesley, Reading, Massachusets, 1995.
9. Сорокин А.С. Подход к моделированию характеристик на основе понятий эквивалентности. // Вестн. Кузбасского гос. тех. унив., 2011. № 1 . С. 128-136.
10. Сорокин А.С. Подход к моделированию характеристик на основе строгих биподобия и бимоделирования.// Вестн. Кузбасского гос. тех. унив., 2009. № 6 . С. 107-119.
11. Сорокин А.С. Моделирование характеристик на основе строгих биподобия и бимоделирования.// Моделирование, программное обеспечение и научноемкие технологии в металлургии. Тр. Третьей всероссийской конф. Новокузнецк, 2011. С. 140-145.
12. Сорокин А.С. Моделирование характеристик на основе отношений между строгим биподобием и марковским процессом.// Моделирование, программное обеспечение и научноемкие технологии в металлургии. Тр. Третьей всероссийской конф. Новокузнецк, 2011. С. 146-149.
13. Малышев В.А. О математических моделях сетей обслуживания// Автоматика и телемеханика. 2009. №12. С. 9-15.
14. Sanders W.H., Meyer J.F. Reduced base model construction methods for stochastic activity networks. //IEEE Journal on Selected Areas in Communications, 9(1), January 1991.
15. Schweitzer P. A Survey of Aggregation-Disaggregation in Large Markov Chains. In W.J. Stewart, editor.//Numerical Solution of Markov Processes, chapter 4,. Marcel Dekker, 1990.
16. Kleinrock L. Queueing Systems, Volume I: Theory. John Wiley, New York, 1975.
17. Baskett F. , Chandy K.M., Muntz R.R., Palacios F.G. Open, Closed and Mixed Networks of Queues with Different Classes of Customers.//Journal of the ACM, 22(2), April 1975.
18. Stifakis J. Use of Petri Nets for Performance Evaluation. In H. Beilner and E. Gelenbe, editors, Measuring, Modelling and Evaluating Computer Systems,. North-Holland, 1977.
19. Molloy M.K. Performance analysis using stochastic petri nets.//IEEE Transactions on Computers, 31(9),1982.
20. Marsan M. A., Conte G., Balbo G. A Class of Generalised Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems.//ACM Transactions on Computer Systems, 2(2), May 1984.
21. Molloy M.K. On the Integration of Delay and Throughput Measures in Distributed Processing Models.//PhD thesis, University of California, Los Angeles, 1981.
22. Henderson W., Taylor P.G. Embedded Processes in Stochastic Petri Nets.// IEEE Transactions on Software Engineering, 17(2), February 1991.
23. Balbo G., Bruell S.C., Ghanta S. Combining Queueing Network and Generalized Stochastic Petri Net Models for the Analysis of some Software Blocking Phenomena.//IEEE Transactions on Software Engineering, 12(4), 1986.
24. Chiola G., Donatelli S., Franceschinis G. GSPNs versus SPNs: What is the actual role of immediate transitions? In Petri Nets and Performance Models, PNPM, Melbourne, Australia, December 1991.
25. Movaghfar A., Meyer J.F. Performability Modelling with Stochastic Activity Networks. //In Proc. of 1984 Real-Time Symposium, Austin, Texas., December 1984.
26. Meyer J.F., Movaghfar A., Sanders W.H. Stochastic activity networks: Structure, behavior and application. In Proc of Int. Workshop on Timed Petri Nets, Torino, Italy., 1985. //IEEE Computer Society Press.
27. Milner R. Communication and Concurrency. Prentice - Hall, 1989.
28. Hoare C.A.R. Communicating Sequential Processes. Prentice-Hall, 1985.
29. Bergstra J.A. , Klop J.W. Algebra for Communicating Processes with Abstraction.//Journal of Theoretical Computer Science, 37, 1985.
30. Plotkin G.D. A Structured Approach to Operational Semantics. Technical Report DAIMI FM-19, //Computer Science Department, Aarhus University, 1981.
31. Milner R. Calculi for synchrony and asynchrony. //Theoretical Computer Science, 25(3), 1983.
32. Moller F., Tofts C. A Temporal Calculus for Communicating Systems. In J.C.M. Baeten and J.W. Klop, editors, CONCUR'90, volume 458 of LNCS. Springer-Verlag, August 1989.
33. Baeten J., Bergstra J. Real Time Process Algebra. Formal Aspects of Computing, 3(2), 1991.
34. Jou C-C., Smolka S.A. Equivalences, Congruences and Complete Axiomatizations of Probabilistic Processes. In J.C.M. Baeten and J.W. Klop, editors, CONCUR'90, volume 458 of LNCS. Springer-Verlag, August 1990.
35. Tofts C. Describing Social Insect Behaviour Using Process Algebra. //Transactions of the Society for Computer Simulation, 9(4), December 1992.
36. Larsen K., Skou A. Bisimulation through Probabilistic Testing.//Information and Computation, 94(1), 1991.
37. Christo L. Specification and Verification Methods for Probabilistic Process. //PhD thesis, Department of Computer Science, Uppsala University, March 1993.

□Автор статьи:

Сорокин
Андрей Семенович,
канд. физ.-мат. наук, доц.,
стар.научн. сотр.
(Филиал КузГТУ, г. Новокузнецк).
Тел.: 8(3843) 772459