УДК 681.5.013

ИНФОРМАЦИОННАЯ СЕТЬ ПЕТРИ КАК ИНСТРУМЕНТ ДЛЯ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ АЛГОРИТМОВ УПРАВЛЕНИЯ

В.А. ИГНАТЕНКО
В.З. МАГЕРГУТ

Белгородский государственный технологический университет им. В. Г. Шухова

e-mail: bigsom@mail.ru

Рассматривается возможность реализации мультипроцессорного режима обработки данных при помощи языка информационных сетей Петри (ИСП). Этот язык позволяет разделять алгоритм на составные части, которые могут функционировать асинхронно на отдельных вычислительных платформах. Применение распределённых вычислений позволяет увеличить число обрабатываемых операций в единицу времени, тем самым достигается увеличение суммарной производительности системы в целом. В статье приводится критерий оптимального разделения сети на подсети, позволяющего добиться оптимальной производительности мультипроцессорной системы.

Ключевые слова: сеть, граф, позиция, дуга, переход, синхронизация, критерий, разделение.

При проектировании систем управления промышленными объектами, как правило, используют модульные системы цифрового управления. Такой способ организации системы управления предполагает ряд особенностей:

закон управления задаётся в виде алгоритма, который обрабатывается на одной или нескольких вычислительных платформах;

данные передаются в цифровом виде непосредственно от интеллектуальных датчиков или производится мультиплексирование аналоговых сигналов с последующей оцифровкой предназначенными для этого модулями предварительной обработки сигналов;

для несложных систем управления, которые могут быть как независимыми системами, так и локальными контурами более сложных систем автоматизации, обычно применяют малогабаритные промышленные контроллеры, не требующие повышенных требований к окружающей среде и допускающих установку в щитах управления или непосредственно на автоматизируемом оборудовании.

Исходя из этих особенностей, а так же из анализа реализуемых в настоящее время проектов автоматизации, можно выделить ряд задач, которые типичны для систем автоматизации объектов малой и средней степени сложности:

при реализации алгоритма управления необходимо предусмотреть возможность его разбиение на части, которые будут выполняться на различных устройствах (локальных контроллерах, интеллектуальных датчиках, устройствах связи, регистрации и архивирования, диспетчерской системе мониторинга и управления);

при модернизации системы управления или при необходимости расширения её функционала может потребоваться повышение производительности вычислительных платформ путём выбора более сложных и дорогих устройств или путём разделения задач между несколькими менее сложными контроллерами;

для многих технологических процессов требуются системы управления повышенной надёжности, например, для потенциально опасных объектов или объектов, аварийный останов которых невозможен или несёт существенные убытки предприятию;

для повышения удобства обслуживания объекта управления требуется разделение системы управления на сегменты по функциональному назначению с возможностью автономного функционирования.

Для решения этих задач необходимо обеспечить возможность распараллеливания основного алгоритма управления и адаптации его частей для асинхронного функциони-

рования на различных вычислительных платформах. Это позволит понизить требования к процессорной мощности отдельно взятого контроллера, более эффективно строить системы с удалёнными объектами управления, повысить надёжность и удобство эксплуатации путём разделения алгоритма на функционально независимые части.

На сегодняшний день для решения задачи параллельного программирования разработан широкий инструментарий [1]. Большинство языков программирования, реализованных как функционально-законченный инструмент проектирования, для реализации параллельной обработки информации используют понятие программного потока (Alice, Occam, T++). Управление потоками реализуется различными средствами, однако у всех инструментов есть общий принцип: в определённых разработчиком узлах алгоритма производится запуск отдельного потока, а в других узлах производится контроль состояния параллельных потоков.

Данный принцип хорошо подходит для тех случаев, когда алгоритм выполняется на одной вычислительной платформе и для разработчика важно контролировать процесс выполнения и время жизни каждого из параллельных потоков. В любом случае, при переходе от алгоритма к тексту программы от разработчика требуется самостоятельно определять механизм создания и взаимодействия параллельных ветвей программы.

Данный принцип малопригоден для применения в системах промышленной автоматизации, в частности при реализации SCADA-систем. Согласно международному стандарту IEC-61131-3 для программируемых логических контроллеров определены следующие языки программирования: IL(Instruction List), LD(Ladder Diagram), FBD(Function Block Diagram), SFC(Sequential Function Chart), ST(Structured Text). Ни один из этих языков не имеет инструментов реализации многопоточной обработки данных. Для реализации этой функции в системах автоматизации обычно используют «master-slave» запросы между отдельными цифровыми устройствами. В этом случае возникает ряд ограничений:

в алгоритме необходимо предусмотреть механизм отправки запроса, формирование ответа и реализовать механизм реакции на этот ответ, таким образом, процесс обмена информацией между частями алгоритма должен быть полностью детерминирован и задан заранее;

пропускная способность канала связи существенно ограничивает скорость обмена информацией и как следствие снижает эффективность обработки информации системой управления;

алгоритм уже не является цельным, а фактически разбивается на независимые алгоритмы, которые должны быть реализованы отдельно, это существенно снижает наглядность и прозрачность закона управления, реализуемого системой автоматического управления.

Таким образом, выпускаемые в настоящее время SCADA-пакеты не предоставляют разработчику простых и понятных инструментов распараллеливания алгоритма и эффективного распределения вычислительных ресурсов.

Для повышения алгоритмической наглядности программ проектируемой системы управления предлагается использование сетей Петри [2, 3].

В отличие от стандартных языков программирования, в которых последовательность выполнения команд определяется счётчиком команд, предложенные сетевые формы отображения позволяют выделять отдельные контуры, объединённые минимальным количеством связей.

Применение информационной сети Петри позволяет разбивать алгоритм на отдельные части, причём для связи частей необходимо передавать минимальный объём информации между вычислительными платформами. По сравнению с классическими средствами реализации параллелизма обработки информации, применение информационной сети Петри позволяет поместить алгоритм на различных контроллерах. При этом аппаратная реализация самих контроллеров не имеет значения, необходимо, чтобы на данной платформе была запущена исполняемая программа. Алгоритм управления для разработчика остаётся единым, благодаря этому, его наглядность не ухудшается.

Для организации структур параллельной обработки данных наиболее подходит информационная сеть Петри (ИСП), которая представляет собой сеть Петри, в которой с переходами сопоставлены условия, а позициям соответствуют операции [2], отличающаяся от нее тем, что позициям могут так же соответствовать промежуточные переменные, определяемые массой фишки, переходы дополнены двумя типами информационных входов (повышающий, обозначаемый кружком на конце перехода с символом «+» и понижающий, обозначаемый кружком на конце перехода с символом «-») и введена дополнительно информационная дуга, которой могут соединяться только позиции с информационными входами переходов или входы и выходы сети с соответствующими вершинами, и которая помечается определенным коэффициентом усиления, а, кроме того, метки в позициях сети имеют массу, которая изменяется по определенным правилам в зависимости от коэффициентов усиления информационных дуг и порога срабатывания перехода, задаваемого информационными дугами.

Одна из идей применения ИСП является разбиение алгоритма на минимальные функционально неделимые блоки. Такими блоками могут быть структуры, реализующие простые операции (арифметические, логические, хранение/выборка, интегрирование, дифференцирование и др.). Блоки включают в себя элементы ИСП – позиции и переходы, причём, все передающие дуги такого блока начинаются и оканчиваются на вершинах, только этого блока. Необходимым и достаточным условием независимости элементарного блока является неизменность суммарной массы фишек этого блока во время работы сети (1). Формально это условие можно записать в виде:

$$Мбл = \sum_{i=1,n} P_i = const, \qquad (1)$$

где *Мбб* – суммарная масса фишек элементарного блока, n – количество позиций, принадлежащих блоку. Данное условие является основным при выборе способа разделения сети на параллельно обрабатываемые подсети. В качестве примера таких блоков можно привести структуры, реализующие операции хранения/выборки (рис.1.а) и суммирования (рис. 1.б).

Блок хранения/выборки функционирует по следующему принципу: при подаче на вход In2 сигнала «0» производится запоминание сигнала In1 и его дублирование массой фишки в позиции P2. При подаче «1» производится сброс элемента в исходное состояние.

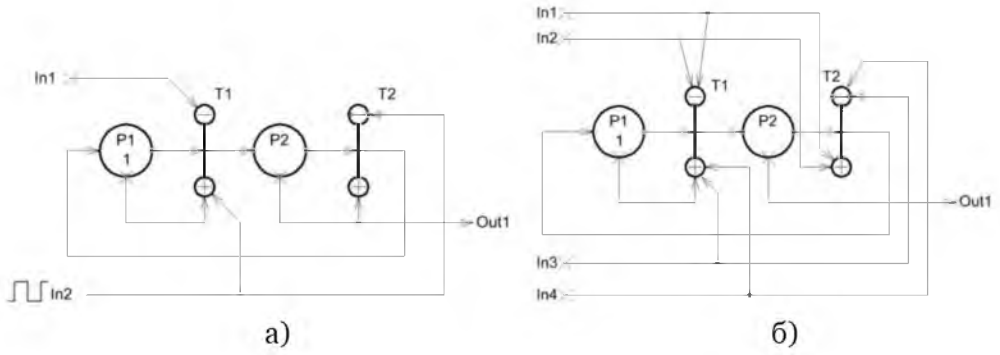

Рис. 1. Примеры элементарных блоков: а – элементарный блок хранения/выборки; б – блок алгебраического суммирования

Операция суммирования, реализуемая элементарным блоком может, формально может быть записана в виде

$$Out1 = In1 + In2 - In3 - In4. \qquad (2)$$

Для произвольного количества входов операцию можно записать в виде:

$$Out1 = \sum_{i=1..n} In_i - \sum_{j=1..m} In_j, \qquad (3)$$

где In_i - положительные входы сумматора, In_j - отрицательные входы сумматора.

Операция выполняется за один проход обработки сети, это означает, что значение Out1 будет вычислено согласно (3) после однократной процедуры переноса массы фишек через переходы T1 и T2 и после этого, при условии неизменности значения входов In1, In2,...,Inn , перераспределение масс фишек по позициям не происходит.

Так как использование ИСП предполагает написание алгоритма цифровой обработки и аналоговой информации (а это вторая идея применения ИСП), то важным вопросом, возникающим при решении такого класса задач является обеспечение достоверности цифрового представления аналоговых сигналов. Для предотвращения потери информации согласно теореме Котельникова-Шенона частота дискретизации аналогового сигнала должна быть не менее удвоенной частоты самого аналогового сигнала. Под частотой дискретизации в данном случае будем принимать величину 1/T, где T – время обработки сетевой структуры элементарного блока. Отсюда можно получить условия:

$$fcsиг \le 1/(2 \cdot T) = 1/(2 \cdot T_1 \cdot n), \quad (4)$$

$$n \le 1/(2 \cdot T_i \cdot fcsиг), \quad (5)$$

где $fcsиг$ – частота аналогового сигнала, T_i – среднее время, затрачиваемое на обработку состояния одного элемента, n – количество элементов ИСП. Для RISC процессоров, на которых проводилось моделирование, соотношение имело следующий вид:

$$n \le fmmак/(2 \cdot k \cdot fcsиг). \quad (6)$$

В данном случае под k подразумевается количество тактов процессора, требуемых для обработки одной позиции информационной сети Петри. Это число зависит от конкретного исполнения исполняющей программы и на данном этапе k=10±4.

Таким образом, неравенство (5) и, как его частный случай, условие (6) являются вторым условием разбиения исходной сети на отдельные сегменты. Это условие гарантирует, что аналоговый сигнал будет обработан корректно, а при вырождении неравенства в равенство, т.е. при $n \to \max$ вычислительные ресурсы процессорного модуля будут использоваться максимально.

Например, если в качестве объекта управления (ОУ) выступает апериодическое звено с передаточной функцией:

$$W(s) = \frac{k}{Ts+1}. \quad (7)$$

Из (7) находим частоту среда апериодического звена:

$$\omega_{cp} = \frac{k}{T}. \quad (8)$$

Подставляя в условие (5) частоту среза (8), получаем максимальное число элементов ИСП, которое может содержать регулятор объекта управления, описываемого апериодическим звеном, реализованный на одном вычислительном узле:

$$n \le \frac{4 \cdot \pi \cdot T}{k \cdot T_i}, \quad (9)$$

где T – постоянная времени ОУ , k – коэффициент усиления ОУ, T_i – среднее время, затрачиваемое на обработку состояния одного элемента ИСП.

При разбиении сети для выполнения на разных аппаратных платформах целесообразно проводить обработку части алгоритма, обрабатывающего высокочастотный сигнал, на более быстродействующих процессорных модулях, а более инерционные части алгоритма можно обрабатывать на процессорных платформах средней и низкой производительности. Кроме того, для локального повышения производительности алгоритма управления (частей отвечающих за предварительную обработку сигнала) можно выделить минимальное количество элементарных блоков для выполнения на отдельно взятом типовом контроллере, тогда, согласно (4), при уменьшении n повышается рабочий

диапазон частот входного сигнала без перехода на другой тип процессорной платформы с более высокой производительностью.

Разделение сети на сегменты осуществляется путём разрыва информационных дуг. С целью оптимизации скорости обработки информации требуется минимизировать количество разрывов между разделяемыми сегментами сети. В связи с этим линия раздела исходной сети проводится исходя из критерия (10), который следует минимизировать. Это связано с тем, что для синхронизации отдельных сегментов сети необходимо передавать информация по разорванным информационным дугам через внешние каналы связи. Так как скорость передачи данных по физическим линиям, как правило, ограничена, и зависит от таких факторов как длина линии связи, тип протокола, характеристика среды передачи информации, то для ускорения процесса обработки информации необходимо сократить объём передаваемой информации за счёт оптимизации механизма синхронизации данных и путём сокращения количества синхронизируемых величин.

$$I = d \to \min_{V} \Big/ \begin{matrix} n \le 1/(2 \cdot T_i \cdot fccuг) \\ Мбб_j = const, \end{matrix} \qquad (10)$$

где d – число разрываемых информационных дуг, V – объём передаваемой информации. При малом числе блоков и сегментов условие (10) может быть выполнено на интуитивном уровне. При большом размере ИСП его выполнение потребует решения оптимизационной задачи.

Таким образом, минимизация разделяемых информационных дуг является третьим условием разбиения ИСП.

Алгоритм разбиение информационной сети Петри выглядит следующим образом:

В структуре сети выделяются элементарные блоки с выполнением условия (1).

Выбирается способ разбиения сети, отвечающий критерию (10).

Место разрыва информационной дуги помечается тождественно равной парой выход-вход, переменные помещаются в таблицу глобальных переменных, синхронизируемую между аппаратными платформами.

Произведённое разбиение ИСП позволило распараллелить производимые операции обработки данных. Однако добавление дополнительных переменных в таблицу синхронизируемых величин увеличивает нагрузку на канал связи между процессорными модулями. Эффективность мультипроцессорной обработки данных подчиняется закону Амдала [4], согласно которому прирост производительности при добавлении одного вычислительного узла снижается с ростом числа этих узлов. Это связано с тем, что в любом разделяемом алгоритме существуют неделимые последовательности действий (команд), которые не могут быть ускорены за счёт параллельного выполнения. Зависимость производительности от числа вычислительных узлов имеет вид:

$$S_P = \frac{1}{\alpha + \dfrac{1-\alpha}{p}}, \qquad (11)$$

где α – доля последовательных вычислений, p – количество вычислительных узлов.

Для существующих распределённых систем управления технологическими объектами величина α может достигать 0.2. Это связано со сложность протоколов, используемых для организации связи между промышленными контроллерами. Применение ИСП существенно снижает долю последовательных вычислений, что существенно повышает эффективность мультипроцессорной обработки (рис. 2).

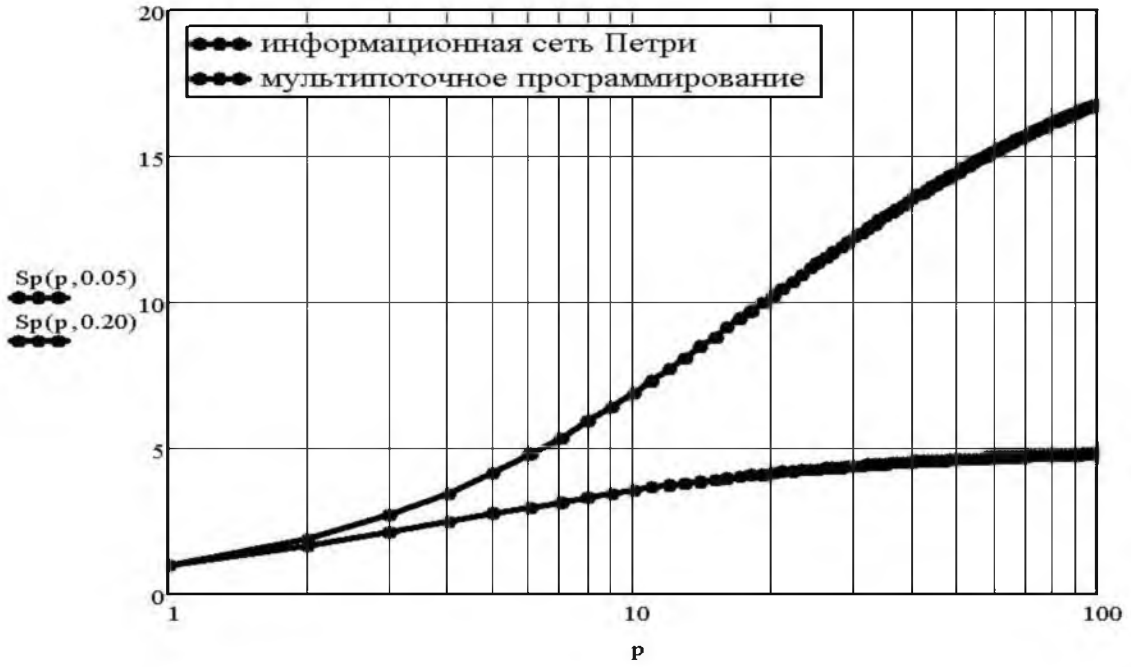

Рис. 2. Эффективность мультипроцессорного режима

Как видно из графика зависимости производительности от количества вычислительных узлов (11) применение ИСП для организации мультипроцессорного режима даёт существенный выигрыш в производительности и увеличивает максимально-достижимую мощность вычислительной системы в целом.

Большая часть последовательных команд связана с процедурой синхронизации таблицы глобальных переменных между процессорными модулями. Определяющее влияние на эту величину оказывает пропускная способность физического канала. Для промышленного применения обычно отдают предпочтение защищённым линиям по сравнению с более быстродействующими, но менее надёжными каналами передачи информации. Поэтому именно качество канала связи будет во многом определять максимальную производительность мультипроцессорной системы автоматического управления промышленными объектами.

Если величину пропускной способности взять постоянной и зависящей лишь от внешних факторов, то эффективность мультипроцессорной обработки зависит от нахождения правильного способа разбиения сети на сегменты. В отличие от классических методов создания параллельных вычислительных структур, где задача разделения процесса вычисления каждый раз решается непосредственно разработчиком и применительно только к текущему проекту, то для алгоритмов, представленных в виде ИСП, процесс разделения сети на подсети может быть формализован и выполнен в автоматическом режиме. Для этого необходимо соблюсти выполнение критерия (10).

В качестве примера разбиения сети ниже приводится узел первичной обработки аналоговой информации, осуществляющий фильтрацию аналоговой величины (рис.3).

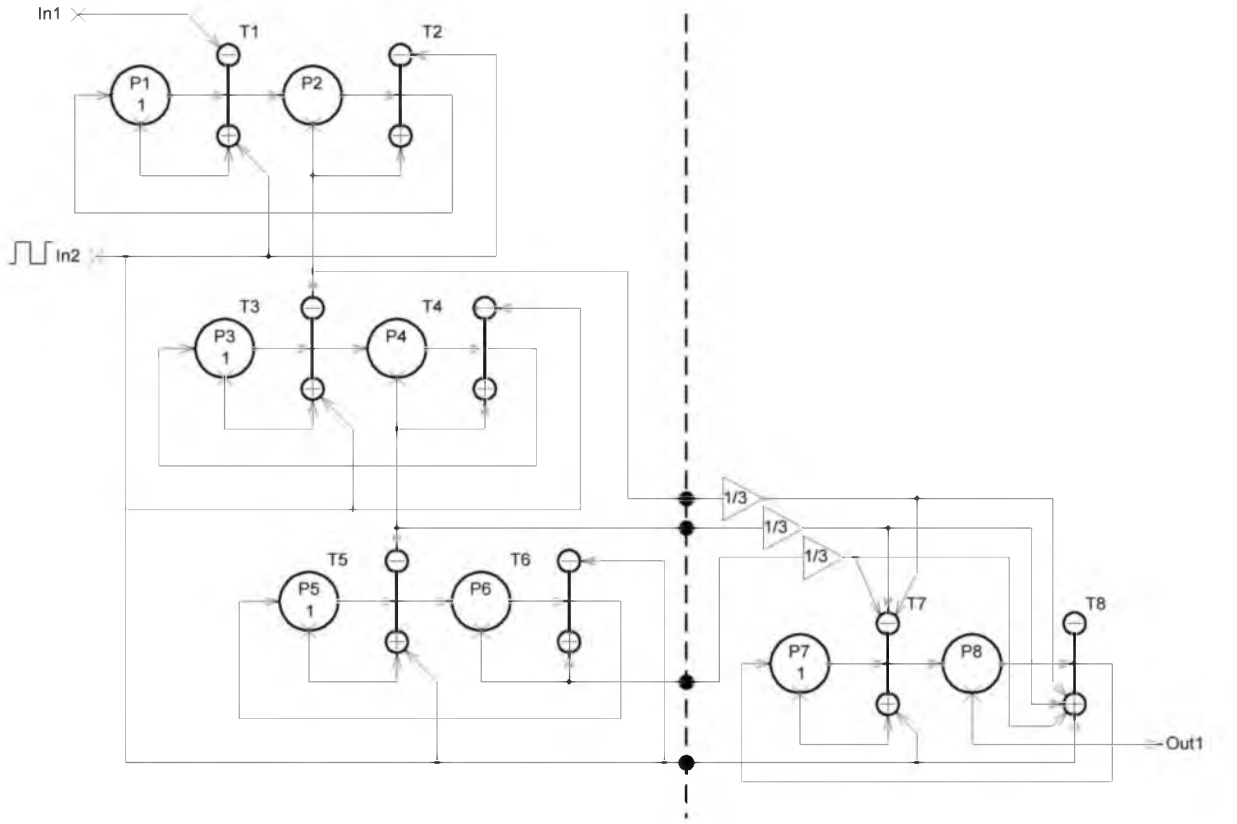

Рис. 3. Алгоритм фильтрации аналогового сигнала

Фильтрация осуществляется по методу скользящего среднего. Алгоритм можно условно разбить на четыре элементарных блока (три блока хранения/выборки, блок суммирования). Линия деления сети пересекает 4 информационные дуги, тем самым образуя 4 дополнительные записи в таблице синхронизируемых переменных.

Таким образом, использование в качестве языка программирования информационной сети Петри для проектирования цифровых систем автоматического управления позволяет реализовать системы с распределёнными вычислениями без снижения наглядности общего алгоритма управления и разработки дополнительного механизма взаимодействия. Разделение сети может быть проведено как в полностью автоматическом режиме, так и с участием человека, который может внести корректировки, связанные с функциональными особенностями разрабатываемого алгоритма управления.

Литература

1. Пратт Т., Зелковиц М. Языки программирования. Разработка и реализация. – М.: Питер, 2002. – 688 с.

2. Юдицкий С.А., Магергут В.З. Логическое управление дискретными процессами. Модели, анализ, синтез. – М.: Машиностроение, 1987. – 176 с.

3. Игнатенко В.А., Магергут В.З. Информационная сеть Петри как основа нечёткого управления объектами. Сб. Трудов XXIII Междун. научн. конф. «Математические методы в технике и технологиях» (ММТТ-23). Т10. Саратов: СГТУ, 2010 – С.13-17.

4. Эрглис К.Э. Интерфейсы открытых систем. – М.: Горячая линия – Телеком, 2000. – 256 с.

INFORMATION NETWORK OF PETRI AS A TOOL FOR PARALLEL PROCESSING CONTROL ALGORITHMS

V.A. IGNATENKO
V.Z. MAGERGUT

Belgorod state technological university named after V.G. Shoukhov

e-mail: bigsom@mail.ru

The possibility of implementing multiprocessor data-processing mode using the language information of Petri nets (ISP). This language allows to divide the algorithm into its component parts, which can operate asynchronously on separate computing platforms. The use of distributed computing increases the number of transactions processed per unit time, thereby achieving an increase in overall system performance in general. In this paper we present a criterion for the optimal partition a network subnet, which allows optimal performance multiprocessor systems.

Key words: network, graph, position, arch, transition, synchronization, criterion, partition.