

ГРУППОВОЙ КОНТРОЛЬ ИНФОРМАЦИОННЫХ ПОТОКОВ В СЛОЖНЫХ СИСТЕМАХ

САЕНКО В.И., КЛИМЕНКО А.В.

Обсуждаются вопросы реализации процедур контроля состояния элементов в организационных информационных системах с административным управлением при обслуживании заявок. Предлагаются конкретные пути реализации процедур контроля для систем различного типа.

1. Актуальность

Одна из проблем, присутствующих в информационных системах, – своевременная обработка потоков заданий, поступающих исполнителям. Из-за высокой интенсивности часть потоков либо обрабатывается с опозданием, либо вообще не обрабатывается, образуя при этом узкие места. Обнаружить эти места в текущий момент времени невозможно, так как они непостоянны и меняются. Однако их удастся выявить путем контроля всех потоков за достаточно длительный период времени. Качество обработки заявок определяет действенность управления всей системой в целом. Степень обработки заявок во многом зависит от характеристик обрабатывающего элемента, поэтому представляется актуальным разработка процедур контроля состояния заявок в системе и процедуры эмпирического определения характеристик элементов [1].

2. Постановка проблемы

Эффективность функционирования информационной системы в значительной мере определяется эффективностью реализации управляющих воздействий. В свою очередь, управляющие воздействия представляют собой задания для исполнителей, поэтому качество обработки этих заданий определяет качество функционирования всей системы и связано с управлением системой обслуживания.

Пусть система обслуживания описывается элементами источников $I = \{i_i; i=1, \dots, i_0\}$, генерирующими задания $R = \{r_k; k=1, \dots, k_0\}$, и элементами исполнителей $P = \{p_j; j=1, \dots, p_0\}$. Каждое активное задание r_k в любой момент времени принимает строго определенное значение параметра состояния $w(R)$, связанное с периодом обработки.

Считаем, что система находится в устойчивом состоянии, при котором все заданные дисциплины обслуживания выполняются в заданный срок. Срыв сроков по вине отдельных I и P приводит к

дестабилизации системы. Задача состоит в выявлении этих I и P и формировании рекомендаций о замене их или их характеристик.

В общем случае поток заданий генерируется множеством (группой) элементов источников и обрабатывается множеством элементов исполнителей. Мощности множеств элементов источников и элементов исполнителей велики, что создает определенные трудности в контроле распределения потоков заданий. Процедуры группового контроля, предложенные в [1], позволяют накапливать статистические данные по обработке потока заданий, а также выявлять подмножества исполнителей, регулярно не справляющихся с обработкой, и группу источников, генерирующих особо сложные элементы потоков заданий.

Рассматривается класс многомерных дискретных многосвязных систем, функционирование которых сводится к обработке непрерывно-дискретного информационного потока заданий в течение определенного периода.

Процедура контроля состояния реализуется в виде информационно-справочной системы, представленной на рис. 1 и предназначенной для эксплуатации в компьютерной сети с администрированием на NetWare и Windows NT и сетевыми рабочими станциями на базе платформы Intel.



Рис. 1. Структурная схема системы контроля

Информационно-справочная система состоит из трех основных модулей: модуля подготовки данных; модуля контроля состояния объекта; информационно-справочного модуля.

Модуль подготовки данных осуществляет ввод потоков заданий R , создает списки источников I , исполнителей заданий P и для каждого I и P формирует дополнительные определяющие характеристики (свойства, технические характеристики параметров, должность, слу-

жебный адрес и т.п.). Все эти характеристики с помощью специального программного интерфейса вводятся в базу данных двумя способами: по шаблону (для элементов, которые имеют общие дополнительные характеристики) и единичному вводу. Ввод по шаблону позволяет вносить элементы по некоторому эталону, который предварительно формирует сам пользователь, что увеличивает скорость подготовки информации и защищает входной поток от ошибок, допускаемых при вводе. Этот эталон включает в себя все параметры, которые характеризуют вводимые элементы. Единичный ввод позволяет вводить информацию для одного элемента без дублирования. Ввод по шаблону используется для генерации заданий R , у которых содержательная часть одинаковая для нескольких или многих P .

Модуль выдачи справочной информации группирует элементы источников I , исполнителей P и заявок R , а также формирует множества:

- всех заданий g_{kij} , сгенерированных источником i_j ;
- всех заданий g_{kij} , направленных исполнителю p_j ;
- всех заданий g_{kij} с любым состоянием $w(R)$;
- всех P , у которых за период времени T обнаружены задания, имеющие статус $w(R)$.

Сформированные элементы множества выводятся в виде списков на экран монитора или в файл.

Модуль контроля состояния объекта формирует множество W , позволяющее отслеживать в любой момент времени состояние элементов g_{kij} , и определяет принадлежность к элементам I_i и P_j . Он также формирует множество W' , для которого характерны одни и те же источники I' и исполнители P' , образующие так называемые «узкие места» в работе системы.

При эксплуатации информационно-справочной системы необходимо вносить отметку в случае, если элементы g_{kij} принимают значения $w(R)$, равные конечным состояниям.

Все модули приложения выполнены в системе программирования Delphi, а в качестве СУБД используется InterBase.

Рассмотрим реализацию процедур контроля для диспетчерских систем на примере задач контроля исполнения поручений, погрузки и разгрузки грузов, обработки деталей.

3. Задача контроля исполнения поручений

Пусть заданы множества распорядителей I (административный аппарат учреждения), исполнителей P (подразделений) и поручений R (приказы и распоряжения), элементы которого генерируются распорядителями и направляются к исполнителям. Каждый элемент g_{kij} характеризуется временем обработки t_{ij} , за которое P_j исполнитель должен выполнить g_{kij} поручение.

Любое поручение может быть выполнено, а может быть и не выполнено. Кроме того, оно может быть выполнено в срок или же после срока, а также может возникнуть необходимость в отмене его выполнения. Проанализировав возможные ситуации, можно выделить шесть состояний (рис.2), которые могут принимать поручение в произвольный момент времени, а именно:

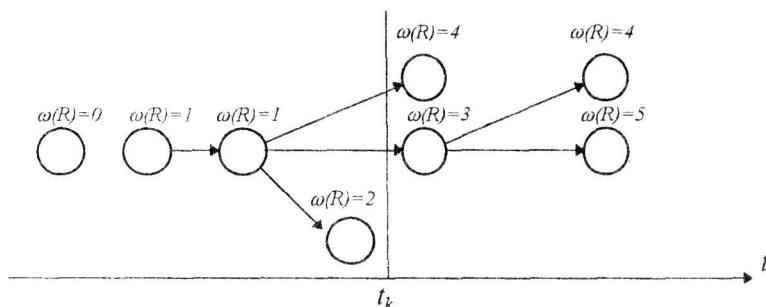


Рис.2. Состояния задачи контроля исполнения поручений

- $w(R) = 0$ – задание Γ_{kij} сформировано, но еще не передано к обработке;
- $w(R) = 1$ – поручения, обрабатываемые в текущий момент времени;
- $w(R) = 2$ – поручения, обработанные к времени t_{ij} ;
- $w(R) = 3$ – поручения, не обработанные к моменту времени t_{ij} и необходимость в обработке которых еще не отпала;
- $w(R) = 4$ – поручения, не обработанные к моменту времени t_{ij} и необходимость в обработке которых уже отпала (старение);
- $w(R) = 5$ – поручения, обработанные позже срока t_{ij} .

Модуль контроля состояния на основании множества состояний $w(R)$, а также множеств источников I и исполнителей поручений P формирует множество W , элементы которого позволяют отслеживать в любой момент времени состояние любого распоряжения Γ_{kij} , а также контролировать состояние (исполнительность) каждого исполнителя или распорядителя. Модуль контроля также формирует подмножество Ω' , включающее в себя распоряжения Γ_{kij} , которые не обработаны ко времени t_k или обработаны позже контрольного срока. На основании элементов подмножества Ω' (просроченных распоряжений) определяются множества исполнителей $I^*(T)$, регулярно не выполняющих свои поручения в течение времени T , и множества распорядителей $P'(T)$, которые генерируют поручения, трудные для выполнения. Изменение характеристик

элементов $I'(T)$ и $P'(T)$ или их замена позволяют повысить качество функционирования всей системы.

4. Задача погрузки и разгрузки грузов

Пусть существуют множества поставщиков I , разгрузочных пунктов P и грузов R , поставляемых i_j поставщиком на p_j разгрузочный пункт. Эту задачу можно представить в виде дискретной системы, в которой есть источники информации (поставщики), потребители информации (разгрузочные пункты) и поток заявок (поток грузов). При работе описанной системы возникают ситуации, препятствующие нормальному режиму работы, такие как затоваривание разгрузочных пунктов, непоставки грузов или поставки с опозданием и т.п. В результате анализа возможных ситуаций выделяем следующие состояния (рис.3):

- t_k – время, к которому должен быть обслужен k -й груз;
- $w(R) = 0$ – груз еще не отправлен;
- $w(R) = 1$ – груз вовремя поступил на разгрузочный пункт и разгружен;
- $w(R) = 2$ – груз вовремя поступил на разгрузочный пункт, но не разгружен;
- $w(R) = 3$ – груз поступил на разгрузочный пункт с опозданием и разгружен;
- $w(R) = 4$ – груз поступил на разгрузочный пункт с опозданием и не разгружен.

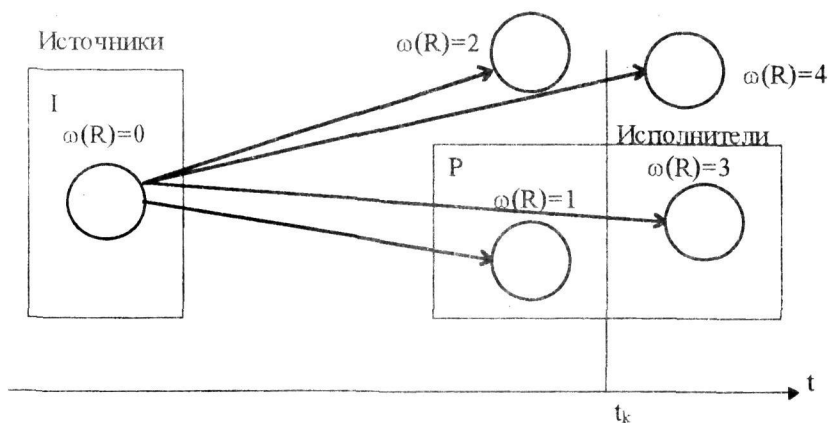


Рис.3. Состояния задачи погрузки и разгрузки грузов

Все элементы информационного потока, которые создают ситуации с $w(R) \in \{0, 3, 4\}$, объединяются в подмножество Ω' . Это

подмножество включает в себя те элементы r_{kij} (грузы), которые поступили на склад позже определенного срока или же вообще не поступили. На основании этих элементов определяются поставщики $I'(t)$, которые регулярно в течение времени t не поставляют грузы или поставляют с опозданием, а также выявляются разгрузочные пункты $P'(t)$, допускающие затоваривание грузов в течение времени T .

5. Задача обработки деталей на множестве станков

Пусть имеется множество деталей I и станков P . Каждая деталь должна быть обработана и поставлена на обработку на следующий станок к определенному моменту времени, иначе могут возникнуть простои станка или его затоваривание деталями. В каждый момент времени принадлежность станка множеству I и P может меняться и определяться только требуемой маршрутной программой для конкретного r_{kij} . Для каждого r_{kij} определяется время t_k , к которому деталь должна быть обработана на станке.

Описанная система будет функционировать нормально (без простоев и затоваривания), если $\Omega' = \emptyset$. А на основании множеств $I'(t)$ и $P'(t)$ можно выявить станки, которые не успевают обрабатывать потоки деталей.

Таким образом, задачи группового контроля дискретной системы имеют широкий спектр применения, например, диспетчерские задачи (контроль материального потока), задачи контроля состояния оборудования (контроль работоспособности элементов оборудования), задачи контроля потока документов.

6. Выводы

Процедуры группового контроля позволяют производить анализ элементов I и P по различным признакам накопленных заданий за выбранный период контроля, который принимает следующие значения: кратковременный контроль (неделя или декада); полный (период исполнения); статистический (месяц); итоговый (1 год или 1 квартал). Процедуры контроля решают задачу группового контроля всех потоков заданий в целях устранения «узких мест». Задача одновременного контроля всех заданий доведена до практической программной реализации.

Предлагаемые процедуры контроля позволяют не только решить локальные задачи мониторинга системы, но и выявить причины понижения эффективности функционирования всей системы.

Литература: 1. Саенко В.И., Клименко А.В. Процедуры группового контроля для информационных систем //Радиоэлектроника и информатика, 1998. №2. С.86-88.

Поступила в редколлегию 28.04.98