

АРХИТЕКТУРА ВРЕМЕНИ В РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

С. П. КОВАЛЕВ

Институт вычислительных технологий СО РАН, Новосибирск, Россия

e-mail: kovalyov@nsc.ru

The general approach to development and implementation of architecture models of distributed information systems is suggested. The approach is based on treatment of internal time of the CSP-system as certain discrete partially ordered set. Various architectural requirements to the system are specified via structural properties of this order: connectivity, consistency, separability etc. Widely used models as client-server and multi-tier architectures are analyzed as examples. Implementation of the system is treated as enclosure of its discrete internal time into continuous real time.

Введение

Современные системы передачи и обработки информации допускают описание без привлечения концепции непрерывного пространства-времени. Действительно, в определениях идеальных объектов, образующих формализацию задач, решаемых на современных компьютерах, не содержится никаких требований по фиксации их относительного расположения в некоторых непрерывных шкалах. В частности, они вовсе не нуждаются в пространстве, поскольку не должны состоять из какой-либо объемлющей среды. Время же достаточно использовать только для придания множеству состояний, проходимых совокупностями этих объектов в процессе решения задач, структуры последовательности [1]. В частности, множество моментов времени информационной системы может быть не более чем счетным.

В качестве элементарной информационной системы рассматривается последовательный процесс — реализация пошагового исполнения некоторого алгоритма над некоторой совокупностью данных. Множество TP моментов внутреннего времени процесса имеет структуру журнала — результата последовательной фиксации каждого очередного состояния процесса на некотором (воображаемом) “внешнем” носителе (в литературе журнал часто называется протоколом [2]). Это множество является изоморфным множеству натуральных чисел ω (либо его некоторому начальному отрезку), поскольку процедура журналирования порождает на нем естественный полный порядок.

Будем рассматривать такие распределенные системы, которые представляют собой совокупности взаимодействующих между собой процессов. Общее (совокупное) время TN любой такой системы строится как прямая сумма:

$$TN = \oplus_i TP_i,$$

где каждое \mathbf{TR}_i есть множество моментов внутреннего времени процесса i . При этом естественный порядок на каждом \mathbf{TR}_i индуцирует лишь частичный порядок на всем \mathbf{TN} . Если процессы остаются независимыми (т. е. фактического взаимодействия не происходит), то структура этого частичного порядка позволяет полностью восстановить разбиение системы на процессы — любые два сравнимых между собой момента времени системы относятся к одному процессу.

Межпроцессное взаимодействие естественным образом приводит к усложнению структуры порядка на множестве моментов времени системы. Детальному изучению этой структуры посвящена данная работа. Важность такого изучения объясняется тем, что, как будет показано далее, характеристические принципы организации межпроцессного взаимодействия, т. е. *архитектурные требования* к системе, могут быть сформулированы как свойства коммуникационного порядка. Иначе говоря, некоторые классы счетных частично упорядоченных множеств могут служить архитектурными моделями для достаточно широкого класса распределенных информационных систем. При реализации таких систем осуществляется вложение их коммуникационного времени в реальное [3].

В настоящей статье сформулирована общая концепция коммуникационного времени, а также рассмотрены некоторые широко распространенные типы его структуры, изложены различные методы разделения произвольных распределенных систем на *архитектурные подсистемы*, реализующие различные типы организации коммуникационного времени. Вводится математический аппарат, формализующий предложенные методы. Далее этот аппарат используется для определения набора архитектурных требований к так называемым многоопорным системам — важному для приложений классу систем. Рассматриваются различные аспекты реализации распределенных систем.

1. Коммуникационное время

Элементарный акт передачи информации от одного процесса к другому состоит из двух событий: отправки порции данных (пакета) процессом-отправителем и ее получения процессом-получателем. При этом каждое из этих двух событий считается неделимым, т. е. занимающим в точности один момент внутреннего времени соответствующего процесса. Поскольку получение может произойти только после отправки, внутренний момент времени процесса-получателя, соответствующий получению данных, должен быть сравним с внутренним моментом времени процесса-отправителя, соответствующим отправке данных, и превосходить его. Таким образом, фиксированная направленность любого акта передачи данных порождает *причинность*, требующую проинтерпретировать регламент доступа к данным, распространяющимся по системе, как общее *коммуникационное время*.

Такая интерпретация согласуется с естественным порядком на внутреннем времени любого процесса: в соответствии с фундаментальными принципами работы с данными для любого алгоритма чтение из любого хранилища возможно только после записи в него. Так что взаимодействие процессов соответствует некоторому, в общем случае достаточно сложному, частичному порядку на общем множестве \mathbf{TN} моментов времени системы, продолжающему естественный индуцированный порядок.

Как отмечалось во введении, естественный порядок на множестве моментов времени каждого процесса является полным. Обобщением свойства полноты на произвольное частично упорядоченное множество является свойство *фундированности* — существование минимального элемента у любого подмножества этого множества. Требование фун-

дированности любого коммуникационного порядка следует из принципа причинности в формулировке аристотелевского первоначала: любая последовательность актов передачи данных должна была когда-либо начаться. Следовательно, изучение архитектуры коммуникационного времени сводится к изучению класса фундированных множеств. Аналогичная связь прослеживается между традиционной теорией последовательных алгоритмов и теорией вполне упорядоченных множеств (ординалов).

При построении первых компьютерных сетей их основной задачей считалось обеспечение способности любого процесса в любой момент его внутреннего времени к отправке данных любому другому процессу и получению данных от любого другого процесса. Идеальными казались такие архитектуры сетей, в которых коммуникационный порядок на **TN** является линейным, так что существует способ реализовать любой акт передачи данных, не противоречащий причинности. В этом случае все процессы, исполняющиеся в различных узлах сети, создают единое одноранговое сетевое время.

Такая архитектура адекватна для открытой сети, предназначенной для максимально широкого обмена информацией посредством диалогов отправителей с получателями. Именно так мыслилось предназначение сети Интернет в начале 80-х годов XX века, от которых осталась еще и поныне иногда употребляющаяся концепция сети как одного большого компьютера. Однако такая идеальная сеть не может справиться с решением сложных практических задач. Действительно, линейный порядок на фундированном множестве **TN** по определению является полным — изоморфным некоторому ординальному числу. Следовательно, некоторый бесконечный начальный отрезок **TN** является изоморфным множеству натуральных чисел ω , так что вся сеть сколь угодно долго ведет себя как единый процесс, последовательно перерабатывающий общий массив данных всех исходных процессов. Даже за пределами этого отрезка не имеется никаких условий взаимной изолированности (отделимости) процессов — они фактически утрачивают аутентичность. С другой стороны, поскольку каждый данный момент коммуникационного времени сети принадлежит внутреннему времени некоторого конкретного процесса, ресурсы сети используются наименее эффективно — в любой момент в точности один процесс активно исполняется, а остальные ждут. Поэтому по мере появления потребности в распределенном информационном обслуживании различных областей человеческой деятельности стали разрабатываться технологии распараллеливания работы информационных систем, обеспечивающие взаимную независимость и защиту процессов и совокупностей взаимодействующих процессов, разделяющих общую среду-носитель актов передачи данных.

Из аксиомы выбора [4] следует, что любой частичный порядок может быть продолжен до линейного (теорема, известная как принцип линейного упорядочения). Несмотря на то, что такое продолжение для произвольного фундированного множества не гарантирует его превращение во вполне упорядоченное, существует достаточно широкий класс сетей (в частности, содержащий любую сеть, поддерживающую конечное число процессов либо существующую в течение конечного интервала реального времени), которые при необходимости могут быть преобразованы в открытые. Для заданного частичного порядка это продолжение может быть неединственным и выбор конкретного расширения определяется балансом желаний (задач) и возможностей разработчиков сети. Например, прямую сумму двух множеств натуральных чисел $\omega_1 \oplus \omega_2$ можно линейно упорядочить по построению, положив любое число из ω_2 превосходящим любое число из ω_1 (при этом получается ординальная сумма $\omega \odot 2$). А можно выполнить совмещение “с половинным сдвигом”, построив множество $\{1_1, 1_2, 2_1, 2_2, \dots\}$. Проинтерпретируем каждое ω_i ($i = 1, 2$) как внутреннее время процесса некоторой двухпроцессной системы. Тогда один (очевидно, наихудший)

способ продолжения порядка описывает вообще не взаимодействующие процессы: первый процесс может передать данные второму только “посмертно” (системы, поддерживающие такие взаимодействия между бесконечными процессами, относятся к классу *зеноновских* [5]). Другой способ соответствует строго *диалоговой* дисциплине взаимодействия, в которой каждый процесс отправляет очередной пакет данных только после получения и обработки пакета-ответа на предыдущий акт и после отправки немедленно переходит в состояние ожидания следующего ответа. Существенно, что диалоговая дисциплина асимметрична — первый акт передачи данных все равно направлен от первого процесса ко второму. Чтобы подчеркнуть эту асимметрию, используют другое название диалоговой дисциплины — *архитектура клиент-сервер*; первый процесс, инициализирующий сеанс взаимодействия, называется клиентом, а второй, запускающийся в состоянии готовности к получению, — сервером. Следует подчеркнуть, что фундаментальность (необычайно широкая применимость) архитектуры клиент-сервер в качестве дисциплины межпроцессных взаимодействий связана с тем, что она задает *изоморфизм* между $\omega_1 \oplus \omega_2$ и ω , т. е. фактически превращает всю двухпроцессную систему (а не только ее начальный отрезок) в одну программу, обеспечивая применимость к ней традиционной теории последовательных алгоритмов.

2. Архитектурные подсистемы распределенных систем

Архитектура коммуникационного времени оказывается особенно важной при построении сред-носителей распределенных вычислений. Многообразие вычислительных алгоритмов исключает возможность свести коммуникационную дисциплину, требующуюся для решения практических задач, к архитектуре клиент-сервер или ее открытым расширениям. Например, состояние отдельных объектов может изменяться по запросам от нескольких источников при условии отсутствия какой-либо корреляции между запросами, а потребность в асинхронном исполнении процедур порождает очень жесткие требования одновременно к отказоустойчивости процессов-заказчиков и изолированности процессов-исполнителей такого режима. В результате структура коммуникационного порядка может быть весьма нетривиальной. Для решения различных задач анализа и проектирования распределенных систем требуется выявлять их основные *архитектурные подсистемы*.

Прежде всего любая распределенная система может оказаться объединением нескольких вообще *не взаимодействующих* между собой подсистем. Например, два автономных компьютера, не связанных общей вычислительной сетью, могут считаться компонентами единой распределенной системы, однако процессы, исполняющиеся на одном из них, никак не взаимодействуют с процессами, исполняющимися на другом. Как отмечалось во введении, коммуникационное время такой системы распадается на совокупность “вещей в себе” — изолированных подсистем, каждая из которых не отправляет никакой информации во внешний мир и не получает ничего извне. Соответственно, архитектурная модель такой системы состоит из взаимно независимых моделей каждой подсистемы.

Другим важнейшим типом архитектурных подсистем любой распределенной системы является *последовательная подсистема* — максимальный отрезок жизненного цикла системы, на котором она ведет себя как изолированная последовательная программа. Очевидным примером последовательной подсистемы является процесс, однако, как было показано в предыдущей главе, некоторые многопроцессные системы также могут вести себя как единый последовательный процесс. Таким образом, последовательная подсистема

определяет максимальную область применимости теории последовательных алгоритмов для изучения данной распределенной системы.

Как будет доказано в следующем разделе, любая распределенная система разбивается на совокупность попарно непересекающихся последовательных подсистем. Поэтому жизненный цикл любого процесса состоит в последовательном прохождении некоторой совокупности последовательных подсистем. Иначе говоря, процесс — это объединение подмножеств линейно упорядоченного множества последовательных подсистем, не пересекающееся с другими процессами, и, кроме того, конечное либо изоморфное ω . Следовательно, если процесс содержит бесконечное множество моментов некоторой последовательной подсистемы, то она должна быть максимальной в нем. Поэтому любой бесконечный процесс либо заканчивается в бесконечной последовательной подсистеме (заиклиивается), либо состоит из счетного множества конечных подмножеств некоторых последовательных подсистем. Примером последнего случая может служить *маршрутизатор* — процесс, задача которого заключается в бесконечной пересылке пакетов данных из некоторого набора источников в некоторый набор назначений, определяемый в результате работы некоторого алгоритма (маршрутизационного решения). Потребность в бесконечных процессах, в принципе не ориентированных на достижение некоторой финальной цели и последующее завершение, является важнейшим отличием распределенных систем от последовательных.

Разбиение распределенной системы на последовательные подсистемы позволяет сформулировать идею гомоморфизма распределенных систем. Две системы называются *гомоморфными*, если их множества последовательных подсистем изоморфны как частично упорядоченные множества. При этом конечные последовательные подсистемы могут переходить в бесконечные и наоборот, поэтому гомоморфные системы могут иметь различные классы возможных разбиений на процессы, однако их структуры межподсистемных взаимодействий идентичны. Этот факт обосновывает выбор разработчиками различных стандартов поддержки распределенных вычислений (RPC, CORBA и т. д.) *процедуры* (метода), а не процесса в качестве минимального программного фрагмента, который может исполняться удаленно. Дело в том, что при реализации распределенных систем на алгоритмических языках программирования именно процедура оказывается языковой сущностью, которая в точности воспроизводит характеристические свойства последовательной подсистемы. А именно, регламент исполнения операторов процедуры фиксирует линейный порядок ее времени и конечность интервала между любыми двумя его моментами, а требование единственности точки входа в процедуру определяет полноту этого порядка.

Выше введены два “предельных” разбиения коммуникационного порядка на подсистемы, выделяющие независимые и последовательные подсистемы. Промежуточные случаи подсистем “общего положения” не могут быть охарактеризованы с такой же точностью. Одной из важнейших аутентификационных характеристик подсистемы является *асинхронность* — возможность работать независимо (по времени) от своего окружения. Асинхронность обеспечивается включением в подсистему *максимальных путей* — последовательностей актов передачи данных, обеспечивающих “эстафетную” передачу от своего начала к концу. Таким образом, асинхронная подсистема содержит некоторые максимальные пути между любыми своими моментами коммуникационного времени, причем эти пути являются достаточно “короткими”, так что вдоль них действительно можно передавать данные. В частности, любая последовательная подсистема асинхронна, так как она представляет собой единый максимальный путь. Любая независимая подсистема также, очевидно, асинхронна. Асинхронная подсистема никогда не впадает в бесконечное ожидание — она всегда работает в соответствии со своей спецификацией, вне зависимости от факта прихода

данных извне подсистемы. В частности, она защищена от потерь данных в коммуникационной среде — например, она может обнаружить, что ожидаемый пакет данных слишком долго не приходит, и попросить подсистему-партнера по взаимодействию повторить его пересылку.

В следующем разделе будут предложены формальные правила построения максимальных путей и выделения асинхронных подсистем. Будет доказано, что взаимодействие любых компонентов любой асинхронной подсистемы поддерживается конечным числом критических *узловых точек*, через которые проходят “почти все” данные, передаваемые между компонентами подсистемы. Такие критические точки, от надежной работы которых зависит все функционирование системы, оказываются, в частности, принадлежащими различным маршрутизаторам и диспетчерам запросов. Идентификация и определение дисциплины функционирования процессов, содержащих узловые точки, являются важнейшими задачами, возникающими при анализе и проектировании архитектур распределенных систем.

3. Математические методы выделения архитектурных подсистем

С формальной точки зрения коммуникационное время представляет собой пару $(\mathbf{TN}, <)$, состоящую из конечного или счетного множества с заданным на нем отношением порядка, причем этот порядок является строгим — он представляет собой иррефлексивное транзитивное бинарное отношение.

Прежде всего научимся выявлять независимые подсистемы, объединением которых может быть \mathbf{TN} .

Определение 1. *Отношением сравнимости \sim называется рефлексивное симметричное замыкание отношения порядка:*

$$\forall t, t' \in \mathbf{TN} : t \sim t' \equiv t = t' \vee t < t' \vee t' < t.$$

Непосредственно из определения следует, что для любых двух элементов выполняется одна и только одна из следующих *альтернатив сравнимости*: либо они совпадают, либо один меньше другого, либо они несравнимы. Если два подмножества \mathbf{TN} таковы, что ни один элемент одного из них не сравним ни с одним элементом другого, то с точки зрения структуры порядка они являются полностью независимыми друг от друга. Для выявления независимых подсистем требуется осуществить транзитивное замыкание отношения сравнимости.

Определение 2. *Отношением порядковой связности \sim^∞ называется транзитивное замыкание отношения сравнимости.*

Отношение \sim^∞ является сильнейшим из нетривиальных отношений эквивалентности, порождаемых отношением сравнимости. Очевидно, любой его класс эквивалентности является полностью независимым от любого другого. Если рассмотреть диаграмму Гессе [4] исходного порядка как *неориентированный* граф, то классы эквивалентности отношения \sim^∞ в точности соответствуют компонентам связности этого графа. Это свойство оправдывает название данного отношения.

Для выявления более “тонких” структурных свойств отношения сравнимости требуется отождествить элементы, не различимые посредством него. Эта задача решается для

произвольного бинарного отношения на произвольном множестве \mathbf{Y} посредством факторизации по некоторому специальному отношению эквивалентности.

Определение 3. Эквиотношением $\#\#$ произвольного бинарного отношения $\#$ называется отношение, определяемое следующим образом:

$$\forall t, t' \in \mathbf{Y} : t\#\#t' \equiv \forall t'' \in \mathbf{Y} (t\#t'' \Leftrightarrow t'\#t'') \& (t''\#t \Leftrightarrow t''\#t').$$

Эквиотношение обладает несколькими очевидными, но важными свойствами.

Предложение 1.

(i) Эквиотношение любого бинарного отношения является отношением эквивалентности.

(ii) Пересечение эквиотношения с исходным отношением определяется “степенью рефлексивности” последнего:

$$\forall t, t' \in \mathbf{Y} : t\#\#t' \Rightarrow (t\#t' \Leftrightarrow (t\#\&t'\#\#t')).$$

В частности, эквиотношение усиливает исходное отношение тогда и только тогда, когда последнее рефлексивно.

(iii) Эквиотношение совпадает с исходным отношением тогда и только тогда, когда последнее является отношением эквивалентности. ■

Факторизуем \mathbf{TN} по эквиотношению, порождаемому сравнимостью.

Определение 4. Отношением эквисравнимости \approx называется эквиотношение отношения сравнимости. Симметричность отношения сравнимости сокращает формальное определение:

$$\forall t, t' \in \mathbf{TN} : t \approx t' \equiv \forall t'' \in \mathbf{TN} t \sim t'' \Leftrightarrow t' \sim t''.$$

Из рефлексивности сравнимости следует, что эквисравнимость усиливает ее. Однако она является слишком слабым отношением, чтобы сохранить *порядковую структуру*, т. е. обеспечить возможность перейти к фактор-множеству, не изменив структуру отношения порядка. В качестве примера рассмотрим четырехэлементное множество $\mathbf{W}_4 \equiv \{1, 2, 3, 4\}$ с отношением порядка, имеющим вид двух параллельных путей: $< \equiv \{(1, 2), (2, 4), (1, 3), (3, 4), (1, 4)\}$. Класс эквисравнимости $\{1, 4\}$ несравним с точкой 2, хотя каждый из его элементов сравним с ней.

Итак, требуется факторизация, согласованная с порядковой структурой. Пример \mathbf{W}_4 позволяет предположить, что нужно усилить эквисравнимость двух элементов требованием эквисравнимости элементов соединяющих их путей диаграммы Гессе. Уточним и докажем это предположение.

Напомним, что (симметричным замкнутым) *интервалом* $[t, t']$ называется множество элементов частично упорядоченного множества, находящихся между t и t' , замкнутое ими самими:

$$\forall t, t' \in \mathbf{TN} : [t, t'] \equiv \{t\} \cup \{t'' \in \mathbf{TN} | t < t'' \& t'' < t'\} \cup \{t'' \in \mathbf{TN} | t' < t'' \& t'' < t\} \cup \{t'\}.$$

Очевидно, что интервал является выпуклым множеством — интервал между любыми двумя его точками содержится в нем.

Определение 5. Отношение интервальной эквисравнимости \approx_{\square} задается следующим образом:

$$\forall t, t' \in \mathbf{TN} : t \approx_{\square} t' \equiv \forall t_1, t_2 \in [t, t'] t_1 \approx t_2.$$

Предложение 2.

- (i) Интервальная эквисравнимость является отношением эквивалентности.
- (ii) Интервальная эквисравнимость усиливает эквисравнимость.
- (iii) Любой класс интервальной эквисравнимости является выпуклым множеством.

Доказательство.

(i) Рефлексивность и симметричность отношения интервальной эквисравнимости очевидны из определения. Транзитивность следует из того, что для всяких t_1, t_2 и t_3 , таких, что $t_1 \approx_{\square} t_2$ и $t_2 \approx_{\square} t_3$, любой элемент из интервала $[t_1, t_3]$ сравним с t_2 , поэтому он попадает либо в $[t_1, t_2]$, либо в $[t_2, t_3]$.

(ii) Очевидно.

(iii) Очевидно, что если два элемента интервально эквисравнимы, то любой элемент из интервала между ними интервально эквисравним с обоими. ■

Докажем, что интервальная эквисравнимость является слабейшим отношением эквивалентности, которое сохраняет порядковую структуру **TN** и согласуется с отношением сравнимости.

Теорема 1. *Любые два интервально неэквисравнимых элемента относятся друг к другу так же, как их классы интервальной эквисравнимости, причем всякое отношение эквивалентности, обладающее этим свойством и не являющееся усилением интервальной эквисравнимости, отождествляет некоторые несравнимые элементы.*

Доказательство. Последнее утверждение доказывается разбором случаев: если некоторое отношение эквивалентности \approx_0 не усиливает интервальную эквисравнимость, то для любого t , класс эквивалентности которого $\approx_0 \langle t \rangle$ не совпадает с классом интервальной эквисравнимости $\approx_{\square} \langle t \rangle$, и для любого $t' \in \approx_0 \langle t \rangle \setminus \approx_{\square} \langle t \rangle$ существует $t'' \in [t, t']$, неэквисравнимый с t . Если $t'' \notin \approx_0 \langle t \rangle$, то $\approx_0 \langle t \rangle$ оказывается несравним с t'' , несмотря на то, что его элементы t и t' сравнимы с t'' . В противном случае существует t^* , сравнимый в точности с одним из t, t'' ; случай $t^* \in \approx_0 \langle t \rangle$ противоречит согласованности со сравнимостью, а случай $t^* \notin \approx_0 \langle t \rangle$ — сохранению порядковой структуры.

Далее, для несравнимых элементов несравнимость их классов интервальной эквисравнимости непосредственно следует из определения эквисравнимости; остается доказать, что если один из интервально неэквисравнимых элементов меньше другого, то его класс интервальной эквисравнимости также меньше класса интервальной эквисравнимости другого:

$$\forall t, t' \in \mathbf{TN} : (t < t' \& \neg t \approx_{\square} t') \Rightarrow (\forall t_1, t_2 \in \mathbf{TN} t_1 \approx_{\square} t \& t_2 \approx_{\square} t' \Rightarrow t_1 < t_2).$$

Достаточно доказать это утверждение при фиксированном $t_2 = t'$. В этом случае из $t \sim t'$ следует, что всегда $t_1 \sim t'$. Но если $t' < t_1$ или $t' = t_1$, то $t' \in [t, t_1]$, поэтому t' оказывается интервально эквисравнимым с t , что противоречит его выбору. Поэтому всегда $t_1 < t'$. Сохранение порядковой структуры отношением интервальной эквисравнимости доказано. ■

Следствие. Если некоторое отношение эквивалентности является усилением отношения интервальной эквисравнимости, причем любой его класс эквивалентности является выпуклым множеством, то оно сохраняет порядковую структуру. ■

Существует другое, более “экономно” проверяемое отношение эквивалентности, совпадающее с интервальной эквисравнимостью. Для его построения достаточно расширить эквисравнимость лишь требованием линейной упорядоченности интервала.

Определение 6. *Отношение линейной эквисравнимости \approx_L задается следующим образом:*

$$\forall t, t' \in \mathbf{TN} : t \approx_L t' \equiv t \approx t' \& \forall t_1, t_2 \in [t, t'] t_1 \sim t_2.$$

Теорема 2. *Линейная эквисравнимость совпадает с интервальной.*

Доказательство. Очевидно, что линейная эквисравнимость ослабляет интервальную. Обратное включение доказывается от противного. Пусть $t \approx_L t'$ и существует $t'' \in [t, t']$, неэквисравнимый с t . Тогда существует t^* , сравнимое в точности с одним из t и t'' . Пусть $\neg t^* \sim t$; тогда если $t^* < t''$, то поскольку $t'' < t'$, транзитивность влечет $t^* < t'$, так что существование такого t^* противоречит эквисравнимости t и t' ; если же $t'' < t^*$, то поскольку $t < t''$, транзитивность влечет $t < t^*$, что противоречит посылке об их несравнимости. Следовательно, t^* сравнимо с t и несравнимо с t'' . При этом если $t^* < t$, то поскольку $t < t''$, транзитивность влечет $t^* < t''$, что противоречит их несравнимости. Значит, справедливо $\neg t^* \sim t'' \& t < t^*$. Определим теперь, как t^* относится к t' . Поскольку $t^* \sim t$, эквисравнимость t и t' влечет $t^* \sim t'$. Если $t' < t^*$ либо $t' = t^*$, то поскольку $t'' < t'$, транзитивность влечет $t'' < t^*$, что противоречит их несравнимости. В итоге имеем $\neg t^* \sim t'' \& t < t^* \& t^* < t'$. Однако конъюнкция двух последних условий означает $t^* \in [t, t']$, а тогда t^* и t'' доставляют опровергающий пример определению линейной эквисравнимости t и t' . Эквивалентность двух отношений доказана. ■

Будем говорить, что отношение эквивалентности сохраняет *ординально-предельную структуру* исходного множества, если любой его класс эквивалентности конечен либо изоморфен ω . Для сохранения ординально-предельной структуры требуется дальнейшее усиление интервальной (линейной) эквисравнимости. Наложим дополнительное требование конечности интервала между любыми двумя эквисравнимыми элементами. Это требование оправдывается следующим фактом.

Предложение 3. Для того чтобы вполне упорядоченное множество было конечным либо изоморфным ω , необходимо и достаточно, чтобы интервал между любыми двумя его элементами был конечным.

Доказательство. Необходимость очевидна, а достаточность следует из того, что любой бесконечный ординал содержит ω и интервал $[0, \omega]$ бесконечен. ■

Для удобства изложения введем предикат конечности $\varphi\mathbf{X}$ частично упорядоченного множества \mathbf{X} согласно

$$\forall \mathbf{X} : \varphi\mathbf{X} \equiv \exists n \in \omega \mathbf{X} = [0, n],$$

где знак равенства означает изоморфизм частично упорядоченных множеств.

Определение 7. *Отношение* полной эквисравнимости \approx_ω *задается следующим образом:*

$$\forall t, t' \in \mathbf{TN} : t \approx_\omega t' \equiv t \approx t' \& \varphi[t, t'].$$

Теорема 3. *Фактор-множество*

$$\mathbf{TS} \equiv \mathbf{TN} / \approx_\omega$$

имеет такую же порядковую и ординально-предельную структуру, что и \mathbf{TN} .

Доказательство. Поскольку истинность предиката конечности влечет линейную упорядоченность интервала, полная эквисравнимость усиливает линейную. А поскольку любой класс полной эквисравнимости является выпуклым множеством, применяем следствие из теоремы 1. ■

Разбиение множества на непересекающиеся подмножества позволяет описывать “жесткие” свойства заданных на нем структур. Другой класс свойств выявляется посредством анализа различных условий *отделимости* подмножеств друг от друга. Рассмотрим одно такое условие.

Напомним, что бинарное отношение $\#^0$ называется *порождающим* данное транзитивное бинарное отношение $\#$, если его транзитивное замыкание совпадает с $\#$. Напомним также, что сужением $\#|_X$ отношения $\#$ на подмножество X называется пересечение $\# \cap X \times X$.

Несмотря на то, что пересечение двух транзитивных отношений транзитивно, операция транзитивного замыкания в общем случае некоммутативна с операцией пересечения. В частности, некоммутативность транзитивного замыкания с сужением иллюстрируется следующим примером. Определим на вышеприведенном множестве W_4 отношение $<^0 \equiv < \setminus (1, 4)$. Очевидно, $<^0$ порождает его отношение порядка $<$. Сужение его на подмножество $X \equiv \{1, 4\}$ пусто, так что остается пустым и транзитивное замыкание этого сужения. Однако сужение исходного отношения $<$ на X содержит пару $(1, 4)$. Можно сказать, что отделение (вычитание) X от W_4 влечет потерю возможности восстановить на X исходное отношение по некоторому порождающему отношению. Оформим эту трактовку в виде определения:

Определение 8. *Подмножество X множества Y называется отделимым (относительно транзитивного отношения $\#$), если сужение на него любого отношения, порождающего данное, коммутативно с транзитивным замыканием:*

$$\forall X \subseteq Y : \delta_{\#} X \equiv \forall \#^0 \subseteq Y \times Y (\#^0)^\infty = \# \Rightarrow (\#^0|_X)^\infty = \#|_X.$$

Поскольку для любого подмножества X справедливо $(\#^0|_X)^\infty \subseteq \#|_X$, для доказательства результатов об отделимости достаточно доказать обратное включение.

Очевидно, что пустое множество и само Y отделимы относительно любого транзитивного отношения. Для выявления других отделимых множеств желательно иметь достаточно “экономный” способ, не требующий перебора всех порождающих отношений. Существует такой способ, основанный на свойствах интервальной структуры множества. Интервал между двумя точками определяется для любого транзитивного отношения точно так же, как для частичного порядка. Будем также говорить, что конечное множество $T_n(\#, t, t')$ связывает две данные точки t и t' некоторым отношением $\#$ (не обязательно транзитивным), если оно содержит ровно $n + 1$ элементов, причем $t_0 = t$, $t_n = t'$, и любая точка, кроме последней, образует со следующей пару, принадлежащую отношению $\#$. Докажем следующий *интервальный критерий отделимости*.

Теорема 4. *Подмножество X отделимо тогда и только тогда, когда для любых двух его точек, входящих в данное отношение, существует связывающее их множество, такое, что интервал между любыми соседними точками из этого множества содержится в X :*

$$\forall X \subseteq Y : \delta_{\#} X \Leftrightarrow (\forall t, t' \in X t \# t' \Leftrightarrow \exists n \in \omega \exists T_n(\#, t, t') : \forall i \in [0, n-1] [t_i, t_{i+1}] \subseteq X).$$

Доказательство. В выражении в скобках обратная импликация следует из транзитивности отношения $\#$, поэтому требуется доказать эквивалентность отделимости и прямой импликации в скобках. В обратную сторону она доказывается непосредственно: по определению транзитивного замыкания, для всякого порождающего отношения $\#^0$ для любой пары $t \# t'$ точек из X существует конечное множество конечных множеств, каждое из которых связывает некоторую пару точек $\{t_i, t_{i+1}\} \subseteq T_n(\#, t, t')$ отношением $\#^0$. Поскольку любая точка каждого из этих множеств содержится в $[t_i, t_{i+1}] \subseteq X$, их объединение является конечным множеством, связывающим t с t' отношением $\#^0|_X$, поэтому $(t, t') \in (\#^0|_X)^\infty$.

Пусть теперь \mathbf{X} — подмножество, для некоторой пары $t\#t'$ которого интервальный критерий не выполнен. Построим трансфинитной индукцией порождающее отношение, противоречащее отделимости \mathbf{X} . Для этого вполне упорядочим некоторым образом множество пар $\mathbf{Y} \times \mathbf{Y}$, так что оно станет изоморфным некоторому ординалу β . Положим $\#_0^0 = \#$. Для любого непердельного ординала $\alpha > \mathbf{0}$ рассмотрим пару $(t_1, t_2)_{\alpha-1}$; если $\{t_1, t_2\} \subseteq \mathbf{X} \cap [t, t']$, $t_1\#t_2$ и вдобавок существует $t^* \in [t_1, t_2] \setminus \mathbf{X}$, то положим $\#_\alpha^0 = \#_{\alpha-1}^0 \setminus (t_1, t_2)_{\alpha-1}$; в противном случае положим $\#_\alpha^0 = \#_{\alpha-1}^0$. Для предельных ординалов α положим $\#_\alpha^0 = \bigcap_{\gamma < \alpha} \#_\gamma^0$. Рассмотрим $\#^0 \equiv \#_{\beta+1}$. Это отношение порождает исходное: оно совпадает с $\#$ для всех пар, принадлежащих $\#$, кроме составленных из точек, содержащихся в $\mathbf{X} \cap [t, t']$, интервал между которыми содержит точку $t^* \notin \mathbf{X}$, а поскольку для таких пар (t_1, t_2) справедливо $t_1\#^0 t^* \& t^* \#^0 t_2$, то $(t_1, t_2) \in (\#^0)^2 \subseteq (\#^0)^\infty$. Однако по построению $\#^0$ любое $\mathbf{T} \equiv \mathbf{T}_n(\#^0, t, t') \subseteq \mathbf{X}$ подходит под интервальный критерий, что противоречит выбору пары (t, t') , следовательно, $(\#^0|_T)^\infty \neq \#|_T$. Интервальный критерий полностью доказан. ■

Следствие 1. Любое выпуклое подмножество отделимо. ■

Следствие 2. Если $\#$ — отношение эквивалентности, то подмножество является отделимым относительно него тогда и только тогда, когда оно является объединением некоторой совокупности классов эквивалентности. ■

Доказательство. Очевидно, что интервалом между любыми двумя эквивалентными точками (в частности, совпадающими) является их общий класс эквивалентности.

Для частично упорядоченного множества \mathbf{TN} отделимость относительно его отношения порядка тесно связана с понятием максимального пути. Рассмотрим множество $\mathbf{TL}[t, t']$ всех путей — линейно упорядоченных подмножеств некоторого интервала $[t, t']$, содержащих его конечные точки. *Максимальным* называется путь, не содержащийся ни в каком другом пути. Существование максимальных путей следует из леммы Цорна [4], поскольку множество путей естественным образом частично упорядочивается отношением включения. Из определения максимального пути следует

Предложение 4. Интервал между любыми двумя точками совпадает с объединением всех максимальных путей, соединяющих эти точки. ■

Будем называть *протопорядком* любое отношение $<^0$, порождающее отношение порядка. Семейство отношений протопорядка, порождающих данный частичный порядок, строится посредством *прореживания*, основывающегося на следующем очевидном факте.

Предложение 5. Если некоторый протопорядок содержит пару, интервал между точками которой содержит еще хотя бы одну точку, то после выбрасывания этой пары отношение остается протопорядком. ■

Если множество \mathbf{TN} фундировано, то для каждой точки любого пути в нем (кроме конца) существует следующая за ним точка этого пути — это минимальный элемент множества всех точек пути, превосходящих данную. Поэтому процесс прореживания для фундированных множеств очевидным образом ограничивается (в трансфинитном смысле) следующим условием.

Предложение 6. Любой протопорядок фундированного множества содержит пару, состоящую из любой точки любого максимального пути и следующей за ним. ■

Можно сказать, что любой протопорядок фундированного множества должен содержать *структуру максимальных путей* порождаемого им порядка, при этом чем сильнее протопорядок, тем меньше другой информации он содержит. В частности, между любыми двумя точками существует конечный максимальный путь тогда и только тогда, когда существует сильнейший протопорядок, содержащий только пары, составленные из точек,

следующих друг за другом в максимальных путях. Именно структура максимальных путей отображается на диаграммах Гессе.

Отделимость подмножеств также выражается как условие на структуру максимальных путей. А именно, предложение 4 позволяет переформулировать интервальный критерий отделимости следующим образом (причем не только для фундированных множеств).

Теорема 5. *Подмножество X частично упорядоченного множества TN является отделимым тогда и только тогда, когда для любых двух его сравнимых точек существует связывающее их множество, такое, что любой максимальный путь, соединяющий эти точки и содержащий это множество, содержится в X . Точки такого множества называются узловыми точками интервала между данными точками.* ■

Следствие 1. Подмножество линейно упорядоченного множества является отделимым тогда и только тогда, когда оно выпукло. ■

Следствие 2. Для того чтобы подмножество было отделимым, необходимо, чтобы оно вместе с любыми двумя сравнимыми точками содержало некоторый соединяющий их максимальный путь. ■

Итак, на любом частично упорядоченном множестве определяются следующие семейства отношений:

$$= \subseteq \approx_{\omega} \subseteq \approx_L \approx \sqcap \subseteq \approx \sim \subseteq \sim^{\infty} \subseteq \times,$$

$$= \subseteq <^0 \subseteq < \subseteq \sim \subseteq \sim^{\infty} \subseteq \times$$

(знак \times обозначает слабейшее из всех бинарных отношений, равное декартовому квадрату исходного множества). Свойства важнейших из этих отношений, в том числе выделяемые ими типы архитектурных подсистем, сводятся в следующую таблицу.

Знак	Наименование	Описываемое структурное свойство	Выделяемые архитектурные подсистемы
\sim^{∞}	Отношение порядковой связности	Сравнимость	Невзаимодействующие подсистемы
\approx_{ω}	Отношение полной эквисравнимости	Порядковая и ординально-предельная структура	Последовательные подсистемы
$<^0$	Семейство отношений протопорядка	Отделимость	Асинхронные подсистемы

4. Многоопорные архитектуры

В качестве примера использования изложенного подхода к проектированию распределенных систем опишем широко распространенную методику распараллеливания приложений, выполненных в архитектуре клиент-сервер — создание *многоопорных архитектур* распределенных систем. Основное преимущество многоопорных архитектур перед традиционной архитектурой клиент-сервер заключается в наличии асинхронных серверов-диспетчеров (application servers), поддерживающих работоспособность всей системы независимо от сбоев, происходящих в сеансах взаимодействия с какими-либо конкретными клиентами.

Соответственно, счетное фундированное порядковое связное множество является моделью некоторой n -опорной архитектуры, если оно состоит из следующих частей:

- $n - 1$ серверов-диспетчеров — отделимых бесконечных процессов $\{\mathbf{TS}_i | i = 1, \dots, n - 1\}$;
- произвольного количества клиентов — совокупностей процессов $\{\mathbf{TC}_i | i = 1, \dots\}$, каждая из которых является инициатором начала и окончания взаимодействия с серверами-диспетчерами, т. е. для каждого клиента $\mathbf{TC} \in \{\mathbf{TC}_i\}$ выполняется условие

$$\exists t_s, t_f \in \mathbf{TC} : \forall t \in \cup_i \mathbf{TS}_i : (\exists t' \in \mathbf{TC} t' \sim t) \Rightarrow t \in [t_s, t_f].$$

Название “ n -опорная” для данной архитектуры оправдывается тем, что в силу (ii) реальные задачи по обработке информации не могут решаться в отсутствие клиентов, поэтому минимальный журнал, описывающий решение реальной задачи, включает моменты времени, принадлежащие по крайней мере n различным процессам.

5. Реализация распределенных систем

Необходимость реализации сред-носителей систем взаимодействующих процессов (в частности, компьютерных сетей) как материальных объектов порождает вложение τ коммуникационного времени в непрерывное реальное:

$$\tau : \mathbf{TN} \rightarrow \mathbf{R}.$$

Очевидно, оно должно быть гомоморфизмом, т. е. согласовывать частичный порядок на \mathbf{TN} с естественным порядком на множестве вещественных чисел. Из теоремы Биркгофа—Мильграма [6] следует, что для любого счетного \mathbf{TN} такое вложение существует. Однако его нельзя трактовать как линеаризацию коммуникационного порядка, поскольку не имеется никакого механизма обеспечения единого отсчета времени для различных компьютеров, даже при их объединении в сеть. Это связано с тем, что информационное время отсчитывается периодическими процессами, происходящими в физическом времени, причем частота передачи сигналов по физической среде сети не может превосходить частоты исполнения процессов подключенного к ней компьютера (иначе он не успевал бы обрабатывать поступающий из сети поток данных). Следовательно, разброс значений физического времени, требующегося для передачи очередного сигнала единого информационного времени, составляет несколько единиц внутреннего информационного времени компьютера, так что синхронизация внутреннего времени по сетевому оказывается невозможной. В лучшем случае можно реализовать статистическую подстройку тактовой частоты относительно частоты некоторого эталонного компьютера в предположении, что отклонение (дисперсия) много меньше ее значения. Такой подстроечный алгоритм лежит в основе службы точного времени сети Интернет, реализованной на семействе прикладных протоколов NTP (Network Time Protocol) [9]. Существенно, что он в принципе не может гарантировать одинаковую частоту ни в какой момент реального времени.

Отображение τ позволяет описать длительность любого акта передачи данных неотрицательным вещественным числом, равным разности моментов реального времени, соответствующих моментам получения и отправки некоторого пакета данных. Неидеальность

физических сред, из которых изготавливаются сети и компьютеры, приводит к невозможности реализовывать акты нулевой длительности. Более того, поскольку во всех физических средах современных сетей пакеты данных кодируются модуляцией сигнала (некоторой характеристики воздействия на субстанцию, из которой сделана сетевая среда), данные могут передаваться только строго последовательно. Поскольку каждый очередной передаваемый бит данных ничем не отличается от предыдущего, длительность любого акта линейно зависит от количества b передаваемой информации:

$$d(b) = kb + d_0.$$

Постоянная составляющая d_0 называется *задержкой*. Коэффициент k задает *скважность* коммуникационного времени относительно реального. Величина, обратная скважности, называется *пропускной способностью* среды.

Другим следствием неидеальности коммуникационной среды является возможность потери передаваемых по ней пакетов данных. Потеря эквивалентна отказу процесса-получателя, однако не существует дисциплины взаимодействия, позволяющей процессу-отправителю заранее определить момент потери и не осуществлять отправку в этот момент. Именно потери не позволяют трактовать даже простейшую систему клиент-сервер как единый процесс. Поэтому измерение и оценка процента потерь очень важны при построении систем взаимодействующих процессов.

Наконец, неидеальность приводит к недетерминированности отображения τ . Иными словами, на самом деле это отображение является *случайным процессом* на множестве моментов коммуникационного времени, причем реальное время, которое, как правило, выступает параметром случайных процессов в естественных науках, здесь само становится случайной величиной. Так что при постановке задачи об изучении компьютерных сетей методами математической статистики происходит скрытая инверсия понятий: детерминированные системы взаимодействующих процессов изучают свое программно-аппаратное окружение, своих пользователей методами математической статистики. Приводившаяся в качестве примера Интернет-служба точного времени NTP является хорошим примером: алгоритм службы включает процедуру статистического оценивания кумулятивной пропускной способности соединения между сервером и клиентом, на основании которой клиент определяет поправку к полученному от сервера значению точного времени, вызванную ненулевой длительностью передачи пакета с этим значением.

Из существования отображения τ для любой структуры коммуникационного времени следует, что материальные среды-носители взаимодействующих процессов не требуют нетривиальных взаимоотношений между системами процессов и пространством. В частности, вся система может размещаться в одной ячейке пространства, разделяя ее ресурсы между отдельными процессами посредством некоторого совмещения со сдвигом в общем коммуникационном времени (по аналогии с архитектурой клиент-сервер). Такой режим функционирования многопроцессных систем широко известен как *разделение времени* (time-sharing).

В случае системы с достаточно большим количеством процессов появляется соблазн воспользоваться отображением τ , чтобы *определить* реальное время через коммуникационное. Иными словами, возникает предположение, что коммуникационный принцип причинности является единственным источником потребности в стреле реального времени. В пользу такого предположения говорят не только отмеченная в предыдущей главе вероятностная природа реального времени “относительно коммуникационного”, но и теоретико-информационные соображения: количество информации равно *неэнтропии* (т. е. энтро-

пии с обратным знаком), так что второе начало термодинамики как будто фиксирует метрическую коллинеарность стрелы времени с потоком информации от отправителя к получателю.

Основными доводами против такого подхода служат принципиальные различия как кардинальных, так и ординальных структур коммуникационного и реального времени: первое является счетным и фундированным, а второе — континуальным и некомпактным. В случае плотного вложения $\text{Im}(\tau)$ в \mathbf{R} можно попробовать осуществить некоторый предельный переход, аналогичный дедекиндову сечению, замыкая “по непрерывности” затравочную систему взаимодействующих процессов (например, алгоритмическую модель статистико-физического газа, представляющего собой множество сталкивающихся и/или дальнодействующих частиц). Альтернативно можно проинтерпретировать дискретность коммуникационного времени в духе квантовой механики, определив физическое наблюдение как акт передачи данных от объекта к субъекту, так что структура коммуникационного порядка точно определит границы физического познания. Наконец, нельзя исключать, что моменты реального времени все же образуют множество меньшей мощности, чем \mathbf{R} . Если какая-либо альтернатива сможет привести к целостной концепции, то в физической картине мира станет еще меньше детерминированности, единственной детерминированной сущностью останется последовательный процесс.

При таком предельном переходе можно, в частности, ожидать, что структура коммуникационного порядка затравочной системы \mathbf{TN} определит *топологическую структуру* на \mathbf{R} . Например, интервалы превратятся в замкнутые множества, а подходящее разбиение \mathbf{TN} на последовательные системы индуцирует на \mathbf{R} структуру многообразия. Можно даже предположить, что некоторые отношения частичного порядка могут порождать многообразия высших размерностей.

Заключение

Полученные в данной работе результаты показывают, что язык описания свойств частично упорядоченных множеств оказывается адекватным как для построения архитектурных моделей распределенных информационных систем, так и для анализа их последующей реализации. Эффективность этого подхода обеспечивается осуществляемой в его рамках формализацией взаимосвязи между временем и взаимодействием. С методологической точки зрения построен дискретный вычислимый аналог теоретико-полевой концепции (пространства-) времени как среды, переносящей взаимодействие.

Список литературы

- [1] TURSKI W. M. Time considered irrelevant for real-time systems // BIT. 1988. Vol. 28. P. 473–476.
- [2] ХОАР Ч. Взаимодействующие последовательные процессы. М.: Мир, 1989.
- [3] ALUR R., HENZINGER T. Logics and models of real time: a survey // Lecture Notes in Computer Science. 1991. Vol. 600. P. 74–106.
- [4] НЕПЕЙВОДА Н. Н. Прикладная логика. Новосибирск: НГУ, 2000.

- [5] МАТНАИ J. Problems, promises and performance: some questions for real-time system specification // Lecture Notes in Computer Science. 1991. Vol. 600. P. 315–324.
- [6] ГОНЧАРОВ С. С., ЕРШОВ Ю. Л., САМОХВАЛОВ К. Ф. Введение в логику и методологию науки. М.: Интерпракс, 1994.
- [7] ВРЕМЯ, хаос и математические проблемы // Тр. сем. МГУ. Вып. 1. М.: МГУ, 1999.
- [8] ГОЛДБЛАТТ Р. Логика времени и вычислимости. М.: ОИЛКРЛ, 1992.
- [9] MILLS D. L. Network Time Protocol (Version 3). RFC1305. 1992.

*Поступила в редакцию 6 марта 2002 г.,
в переработанном виде — 18 июля 2002 г.*