УДК 519.633.9

СРЕДСТВА АВТОМАТИЗАЦИИ СОГЛАСОВАНИЯ КОЛЛЕКЦИЙ ФАЙЛОВ НА АВТОНОМНЫХ КОМПЬЮТЕРАХ

О. Б. Арушанян[1], Н. А. Богомолов[1], А. Д. Ковалев[1]

Рассматриваются проблемы согласования содержимого коллекций файлов, а также подходы к автоматизации репликации изменений коллекций файлов на автономных компьютерах для поддержки работ группы пользователей над общим проектом, в том числе для сопровождения файлов программ и документации Библиотеки численного анализа НИВЦ МГУ.

Ключевые слова: репликация файлов, инструментальные средства, численный анализ, библиотеки программ

В течение ряда лет в НИВЦ МГУ ведутся работы по развитию и сопровождению Библиотеки программ решения типовых задач численного анализа [1, 2]. Библиотека представляет собой коллекцию файлов, содержащую более 15 тысяч файлов с текстами программ на языках Фортран и Си, примерами их использования и описаниями. Для сопровождения этой коллекции весьма актуальной является задача автоматизации процесса согласования (синхронизации) файлов на компьютерах сотрудников. Во многих случаях компьютеры, на которых необходимо согласовывать файлы, не имеют непосредственного доступа к файловой системе друг друга. Такие компьютеры мы будем называть автономными. Задача синхронизации множества файлов на автономных компьютерах состоит в периодическом переносе измененных файлов с компьютера на компьютер таким образом, чтобы множество файлов после синхронизации имело на каждом компьютере согласованный состав и содержимое отдельных файлов. Необходимо учитывать, что на компьютерах пользователей может размещаться не вся совокупность синхронизируемых файлов, а лишь их подмножество, необходимое для работы каждого конкретного пользователя. Различный состав файлов на компьютерах пользователей требует индивидуальной настройки процедур синхронизации для каждой пары участвующих в этом процессе компьютеров. Для минимизации объема информации, переносимой с компьютера на компьютер в процессе репликации изменений, необходимо на компьютере "источнике изменений" (далее компьютер-источник) уметь выделять те файлы, в которые отличаются от соответствующих файлов на компьютере "получателе изменений" (далее компьютер-получатель). Необходимость учета всех перечисленных выше факторов делает задачу согласования файлов не вполне тривиальной — неоднозначной и достаточно "запутанной". Целью авторов работы было создание методики работы пользователей с коллекциями файлов, которая позволила бы формализовать процедуры синхронизации файлов на автономных компьютерах и разработать соответствующие программные средства.

Важным частным случаем согласования файлов, заслуживающим отдельного рассмотрения, является ситуация, в которой на нескольких автономных компьютерах поочередно работает один пользователь. Задача синхронизации в этом случае существенно упрощается, так как изменения файлов могут происходить только на том компьютере, на котором в данный момент работает пользователь. В этой ситуации каждый переход пользователя на очередное рабочее место "естественным" образом определяет и момент времени для выполнения процедуры синхронизации и пару участвующих в синхронизации компьютеров. Анализ согласования файлов при работе одного пользователя на нескольких автономных компьютерах позволяет проиллюстрировать ряд общих механизмов организации процесса синхронизации файлов, которые можно с успехом применять и в случае многих пользователей. Одним из таких механизмов является описанный ниже "механизм граничных дат", используемый на компьютере-источнике для отбора файлов, предназначенных для обновления данных на компьютере-получателе.

Рассмотрим подробнее процедуру согласования файлов в случае поочередной работы одного пользователя на двух компьютерах C_1 и C_2. Пусть начиная с некоторого момента времени T_{10} файлы на компьютерах C_1 и C_2 оказались в согласованном состоянии и после этого пользователь начинает работать на компьютере C_1. Если в некоторый момент времени T_{11} пользователь завершает работу на компьютере C_1 для продолжения работы на компьютере C_2, то для согласования файлов на компьютерах ему необходимо подготовить на компьютере-источнике C_1 "синхронизирующую посылку" SDF_{12} (Synchronized Data

[1] Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, 119991, Москва; О. Б. Арушанян, зав.лабораторией, e-mail: arush@srcc.msu.ru; Н. А. Богомолов, ст. научн. сотр., nbogom@srcc.msu.ru; А. Д. Ковалев, вед. научн. сотр., kovalev@srcc.msu.ru

Files), состоящую из файлов, которые были изменены в интервале времени, начиная с T_{10} до момента подготовки "синхронизирующей посылки" T_{11}. Будем называть момент времени, начиная с которого выполняется "захват" (отбор) файлов для включения в "синхронизирующую посылку", нижней граничной датой синхронизации и обозначать ее далее SLD (Synchronization Low Date), а момент времени начала подготовки "синхронизирующей посылки" — верхней граничной датой синхронизации и обозначать ее далее SUD (Synchronization Upload Date). Файлы, включаемые в такую "синхронизирующую посылку", гарантировано лежат в интервале времени (SLD, SUD).

До начала работы на компьютере C_2 пользователь должен перенести файлы из подготовленной ранее на компьютере C_1 "синхронизирующей посылки" SDF_{12} в соответствующее место на компьютере C_2. После завершения переноса обновлений в момент времени T_{21} ($T_{21} > T_{11}$) файлы на компьютере C_2 приводятся в соответствие с состоянием файлов на компьютере C_1 на момент времени T_{11} (на момент гарантированного завершения изменения файлов на компьютере C_1). При обратном переходе пользователя на компьютер C_1 в момент времени T_{22} необходимо аналогичным образом подготовить "синхронизирующую посылку" SDF_{21} с файлами, измененными пользователем в процессе работы на компьютере C_2. Хотя и кажется вполне естественным выбрать в этом случае момент времени T_{21} в качестве нижней граничной даты для включения файлов в "синхронизирующую посылку" SDF_{21}, более правильным является использование для этой цели момента времени T_{11}, соответствующего предыдущему согласованному состоянию файлов или верхней граничной дате подготовки предыдущей "синхронизирующей посылки" (C_1.SUD). Аналогично при очередной подготовке синхронизирующей посылки на компьютере C_1 в качестве нижней граничной даты надо использовать момент подготовки синхронизирующей посылки на компьютере C_2 (C_2.SUD $= T_{22}$). Таким образом, осуществляется взаимное обновление на паре компьютеров, участвующих в согласовании файлов.

Как мы видим, верхняя граничная дата синхронизации C_s.SUD (момент начала подготовки "синхронизирующей посылки") на компьютере-источнике C_s становится нижней граничной датой синхронизации C_d.SLD при обратной отправке изменений с компьютера-получателя C_d. Для того чтобы на компьютере-получателе можно было воспользоваться информацией о времени начала подготовки "синхронизирующей посылки" на компьютере-источнике, необходимо включить эти данные в "синхронизирующую посылку" и сохранить в компьютере-получателе для дальнейшего использования при подготовке обратной "синхронизирующей посылки" SDF_{sd}.SUD $= C_s$.SUD и далее C_d.SLD $= SDF_{sd}$.SUD.

Помимо верхней граничной даты синхронизации целесообразным оказывается включение в "синхронизирующую посылку" и сведений о ее нижней граничной дате (SDF_{sd}.SLD $= C_s$.SLD).

При правильно организованном взаимном обновлении этот процесс должен быть "непрерывным" и состоять из следующих строго чередующихся "шагов":

1) работа на компьютере C_1,
2) подготовка обновлений SDF_{12},
3) перенос их на компьютер C_2,
4) работа на компьютере C_2,
5) подготовка обновлений SDF_{21},
6) перенос их на компьютер C_1.

Если случайно при переходе с компьютера на компьютер пользователь пропустит "шаги" 2–3 или 3–4, то произойдет рассогласование файлов. Контроль правильности действий пользователя по согласованию файлов можно выполнять в момент внесений обновлений на компьютер-получатель C_d. Индикатором "непрерывности" взаимных обновлений является соотношение между нижней граничной датой "синхронизирующей посылки" SDF_{sd}.SLD и предыдущей верхней граничной датой синхронизации на этом компьютере C_d.SUD. Если они в точности равны, то это означает, что цикл шагов по согласованию файлов не был нарушен. Если SDF_{sd}.SLD больше C_d.SUD, то это позволяет предположить, что какие-то шаги в цикле согласования были пропущены и на компьютере-получателе C_d остались не обновленными файлы в интервале (C_d.SUD, SDF_{sd}.SLD). Отклонение нижней граничной даты "синхронизирующей посылки" в обратную строну (SDF_{sd}.SLD $< C_d$.SUD) указывает на возможную избыточность этой порции обновлений, которая может содержать в интервале (SDF_{sd}.SLD, C_d.SUD) файлы, уже имеющиеся на компьютере C_d.

Во всех предыдущих рассуждениях предполагалось, что изменение содержимого увеличивает дату изменения файла. Однако изменение в форме "отката" к более ранней версии файла уменьшает его дату изменения. Для того чтобы в рамках описанного выше "механизма граничных дат" "захватить" (отобрать) в "синхронизирующую посылку" файлы, для которых был выполнен возврат к более ранней версии, необходимо соответствующим образом уменьшить нижнюю граничную дату синхронизации, которая была сформирована на основании прошлой "синхронизирующей посылки". Уменьшение нижней граничной да-

ты синхронизации приведет к избыточности "синхронизирующей посылки", но поскольку изменение в форме "отката" к более ранней версии файлов — явление достаточно редкое, то избыточность в этом случае можно считать вполне допустимой.

Рассмотренную выше схему согласования коллекций файлов для одного пользователя и двух компьютеров можно применить и для большего числа компьютеров. Общий процесс согласования в этом случае можно представить как последовательную синхронизацию файлов на парах компьютеров в соответствии с "траекторией" перемещения пользователя между компьютерами. "Механизм граничных дат" обеспечивает распространение обновлений файлов вдоль "траектории" перемещения пользователя. Однако реализация "механизма граничных дат" в случае нескольких компьютеров должна учитывать следующее.

Во-первых, поскольку "траектория" перемещения пользователя не известна заранее, то на каждом из N участвующих в синхронизации компьютерах нужно хранить индивидуальные нижние граничные даты для остальных N-1 компьютеров, участвующих в согласовании файлов.

Во-вторых, при выполнении обновления файлов на компьютере-получателе C_d файлами, измененными на компьютере-источнике C_s ("прямое" обновление), может выполняться еще и обновление файлами, которые были ранее перенесены на компьютер C_s за счет предыдущих обновлений с других компьютеров ("косвенное" обновление). Если не учитывать эффект "косвенного" обновления, то при последующем переносе файлов с компьютера C_d на компьютеры-источники этого "косвенного" обновления может возникнуть избыточность.

В-третьих, необходимо ввести специальную коррекцию нижних граничных дат на компьютере-получателе C_d при внесении обновлений, вызванных "откатом" к более ранним версиям файлов на компьютере-источнике C_s, чтобы "механизм граничных дат" обеспечил распространение подобных обновлений и в условиях, когда в согласовании участвуют более двух компьютеров.

Для иллюстрации эффекта "косвенного" обновления рассмотрим следующий пример. Предположим, что на трех компьютерах C_1, C_2 и C_3 файлы находятся в согласованном состоянии на некоторый момент времени T_0. Пусть пользователь начинает работу на компьютере C_1, затем переходит последовательно на C_2 и C_3 в моменты времени T_{12} и T_{23} и вновь возвращается на C_1 в момент времени T_{31}. "Синхронизирующая посылка" SDF_{12}, формируемая при переходе на компьютер C_2, будет содержать файлы, измененные в процессе работы на C_1 в интервале (T_0, T_{12}). Соответственно, "синхронизирующая посылка" SDF_{23}, формируемая при переходе на компьютер C_3, будет содержать файлы, измененные в интервале (T_0, T_{23}). В состав SDF_{23} войдут файлы "прямого" обновления, измененные на C_2 в интервале времени (T_{12}, T_{23}), и файлы "косвенного" обновления, измененные на C_1 в интервале времени (T_0, T_{12}). По аналогии с предыдущими посылками, "синхронизирующая посылка" SDF_{31}, формируемая при переходе на компьютер C_3, должна содержать файлы, измененные в интервале (T_0, T_{31}). В состав подготовленной таким образом посылки SDF_{31} войдут файлы, измененные на C_3 в интервале времени (T_{23}, T_{31}), файлы "косвенного" обновления, измененные на C_2 в интервале времени (T_{12}, T_{23}), и явно избыточные файлы "косвенного" обновления, измененные на C_1 в интервале времени (T_0, T_{12}). Избыточности, вызываемой "косвенным" обновлением, можно избежать за счет своевременной коррекции граничных дат на компьютере-получателе изменений. Для этого в состав "синхронизирующей посылки" SDF_{sd} необходимо включать информацию о состоянии нижних граничных дат для всех участвующих в синхронизации компьютеров, которые установлены на компьютере-источнике C_s в момент подготовки обновлений.

Ниже приведен алгоритм коррекции граничных дат на компьютере-получателе C_d при приеме "синхронизирующей посылки" SDF_{sd}, учитывающий эффекты "косвенного" обновления и обеспечивающий распространение изменений, вызванных "откатом" к более ранним версиям файлов.

```
for  i := 0  to  N-1  do
begin
    if  i = d  then  continue; // d — индекс компьютера-получателя
    // коррекция нижних граничных дат для остальных
    if  i = s  then  // s — индекс компьютера-источника
    // коррекция нижней граничной даты для компьютера-источника
        C.SLD[i] := SDF.ULD;
    else
    begin
    // коррекция нижней граничной даты для остальных компьютеров
    if  SDF.SLD[i] > C.SLD[i]  then
        C.SLD[i] := SDF.SLD[i] // увеличение нижней граничной даты для
                               // исключения избыточности вызванной
```

```
                                    // "косвенными" обновлениями
        else
        if   SDF.SLD < C.SLD[i]   then
             C.SLD[i] := SDF.SLD // уменьшение нижней граничной даты для
                                 // распространения изменений, связанных с "откатом"
                                 // к более ранним версиям файлов
        end;
    end;
```

Описываемые выше правила формирования "синхронизирующей посылки" SDF_{sd} на компьютере-источнике C_s предполагают, что она адресована определенному компьютеру-получателю C_d, поскольку при ее подготовке используется нижняя граничная дата, учитывающая историю обновлений файлов между этими компьютерами. В некоторых случаях может оказаться полезным использование для обновления файлов "синхронизирующей посылки", адресованной другому компьютеру. Оказывается, что иногда это возможно, однако для выполнения такого обновления требуется специальная проверка соотношения граничных дат синхронизации и внесение определенных изменений в приведенный выше алгоритм коррекции граничных дат.

Рассмотрим подробнее условия, при которых прием "чужих" обновлений возможен. Пусть "синхронизирующая" посылка SDF_{sd} подготовлена на компьютере C_s для компьютера C_d, а вносится на компьютере C_n. Если на компьютере-получателе C_n нижняя граничная дата для компьютера-источника $C_n.SLD_s$ меньше, чем нижняя граничная дата обновления "синхронизирующей посылки" $SDF_{sd}.SLD$, то согласование может оказаться неполным, так как на компьютер C_n могут не попасть файлы, измененные на компьютере C_s в интервале времени $(C_n.SLD_s, SDF_{sd}.SLD)$. Если же величина $C_n.SLD_s$ больше или равна $SDF_{sd}.SLD$, то файлы на компьютере-получателе C_n будут согласованы с файлами на компьютере-источнике C_s, но, возможно, обновление SDF_{sd} будет содержать избыточные файлы в интервале $(SDF_{sd}.SLD, C_n.SLD_s)$.

Коррекция граничных дат на компьютере C_n должна выполняться по тому же алгоритму за исключением того, что увеличение граничной даты для компенсации избыточности от "косвенного" обновления не должно выполняться для компьютера C_d — настоящего адресата обновлений.

Теперь вернемся к исходной задаче согласования файлов, когда изменения файлов осуществляются на различных компьютерах несколькими пользователями. Главная сложность согласования в этом случае обусловлена тем, что изменения в коллекции файлов вносятся одновременно на многих компьютерах. Однако если исключить возможность возникновения коллизий, обусловленных "одновременным" редактированием копий одного файла на различных компьютерах (поскольку разрешение этих коллизий практически не поддается автоматизации), то задача согласования файлов становится разрешимой. Для исключения "одновременного" редактирования необходимо разделить все множество синхронизируемых файлов на "непересекающиеся" наборы. Копии экземпляров таких наборов могут находиться на многих компьютерах, но изменения в файлы набора может вносить только один пользователь. Закрепление "ответственности" за изменение содержания определенных файлов за конкретными участниками коллектива не является слишком "сильным" ограничением, препятствующим работе над проектом.

В такой постановке процесс согласования всего множества файлов можно рассматривать как совокупность происходящих независимо друг от друга процессов согласования различных наборов файлов. При таком подходе требуется независимое ведение граничных дат для каждого набора на каждом компьютере. Процесс синхронизации одного такого набора похож на согласование файлов на нескольких компьютерах с одним пользователем. Отличие состоит в том, что, как правило, пользователь работает не на всех участвующих в синхронизации компьютерах (часто на одном). Все изменения файлов одного набора в этом случае распространяются в одну сторону от компьютера, на котором вносятся изменения, к остальным участвующим в согласовании компьютерам. Описанный выше "механизм граничных дат" при однонаправленном распространении изменений будет приводить к избыточности "синхронизирующих посылок". Для исключения такой избыточности необходимо вести коррекцию нижних граничных дат набора файлов не только при внесении обновлений на компьютер-получатель C_d, но и при подготовке обновлений на компьютере-источнике C_s, изменив после завершения подготовки обновлений нижнюю граничную дату набора файлов SLD для компьютера C_d ($C_s.SLD_d$) на верхнюю граничную дату "синхронизирующей посылки" ($SDF_{sd}.SUD$).

Практически процесс согласования всего множества файлов выполняется как множество происходящих независимо друг от друга процессов согласования файлов между различными парами участвующих в синхронизации компьютеров. При подготовке обновлений от компьютера-источника C_s к компьютеру-получателю C_d требуется одновременное согласование нескольких файловых наборов, копии которых

имеются на обоих компьютерах. Для этого достаточно объединить независимо подготавливаемые на компьютере-источнике C_s "синхронизирующие посылки" для каждого файлового набора в общий пакет, а внесение изменений из такого "синхронизирующего пакета" выполнить на компьютере-получателе C_d также независимо для каждого представленного в пакете файлового набора.

Несмотря на закрепление "ответственности" за содержание файлов набора за конкретными пользователями, иногда возникают ситуации, в которых изменения в файлы набора могут вноситься на компьютерах других пользователей, не являющихся "ответственными" за изменение файлов этого набора. В этом случае возникает необходимость организации "встречного" обновления файлов набора. Для реализации выполняемых асинхронно "встречных" обновлений без нарушения согласованности файлов требуется внесение некоторых изменений в алгоритм коррекции граничных дат, которая осуществляется на компьютере-получателе обновлений.

Рассмотрим подробнее ситуацию внесения "синхронизирующей посылки" в случае необходимости выполнения последующего "встречного" обновления. До подготовки "синхронизирующей посылки" изменения в файлы наборов вносились на обоих участвующих в синхронизации компьютерах C_s и C_d. Пусть на компьютере-получателе C_d осуществляется прием обновлений SDF_{sd}, подготовленных на компьютере-источнике C_s. В обычной ситуации после приема обновлений нижняя граничная дата для компьютера C_s ($C_d.SLD_s$) должна замениться на верхнюю граничную дату "синхронизирующей посылки" $SDF_{sd}.SUD$. Однако при таком изменении нижней граничной даты в обратную "синхронизирующую посылку" SDF_{ds} не попадут файлы, изменения которых были сделаны ранее на компьютере C_d. Поэтому для обеспечения согласования файлов необходимо отказаться от такой коррекции нижней граничной даты при внесении обновлений, что в свою очередь приведет к избыточности "синхронизирующих посылок".

Для уменьшения избыточности "синхронизирующих посылок" в условиях потенциального "встречного" обновления возможно следующее решение. Распределение файлов коллекции по наборам, содержимое которых контролируется различными пользователями, фактически изменяет статус соответствующих наборов при их размещении на различных компьютерах. Если файловый набор находится на компьютере, на котором работает пользователь, осуществляющий в основном его изменение, такой набор имеет статус "основной". В противном случае аналогичный файловый набор имеет статус "подчиненный". При правильном распределении файлов коллекции по наборам, содержимое которых контролируются различными пользователями, распространение обновлений происходит, как правило, от "основного" файлового набора к "подчиненному". При обновлении "основного" файлового набора (которое должно происходить достаточно редко) нужно запретить коррекцию нижней граничной даты, как это делается для описанного выше "встречного" обновления. При обновлении "подчиненного" набора можно сохранить алгоритм коррекции нижних граничных дат, исключающий избыточность.

Если необходимо выполнить "встречное" обновление файловых наборов, то надо придерживаться следующей последовательности действий. Сначала требуется выполнить обновление "основного" файлового набора, которое выполняется без коррекции нижней граничной даты. Само это обновление будет выполнено без избыточности. Затем можно выполнить обновление "подчиненного" файлового набора, которое будет выполнено с избыточностью, поскольку в него попадут файлы, полученные ранее из этого "подчиненного" файлового набора.

Предложенные методика работы пользователей с коллекциями файлов, а также правила и алгоритмы согласования этих коллекций позволяют разработать не очень сложные в реализации и достаточно простые в эксплуатации программные средства автоматизации согласования файлов на автономных компьютерах. Эти программные средства могут оказаться практически полезными не только в рамках выполняемых в НИВЦ МГУ работ по сопровождению коллекции файлов Библиотеки численного анализа, но и при выполнении других программных проектов.

СПИСОК ЛИТЕРАТУРЫ

1. *Воеводин В.В., Арушанян О.Б.* Структура и организация Библиотеки численного анализа НИВЦ МГУ // Численный анализ на ФОРТРАНе. Вычислительные методы и инструментальные системы. М.: Изд-во Моск. ун-та, 1979. 73–83.
2. *Арушанян О.Б., Волченскова Н.И.* Библиотека программ НИВЦ МГУ для решения типовых задач численного анализа // Вычислительные методы и программирование. 2002. **3**, № 2. 158–163.

Поступила в редакцию
25.09.2008