Сравнение объектов хранения компонентного каркаса системы Блэкбокс

Е.Э. Темиргалеев

В данной заметке приведены результаты тестирования общего способа проверки на равенство для двух объектов хранения. Метод не зависит от конкретного типа объектов хранения, т. к. заключается в сравнении байтовых последовательностей, получаемых при выгрузке. Для выгрузки используются спец. реализации абстракции файлов.

Основные термины и постановка задачи. *Блэкбокс* — среда выполнения компонентного ПО, компонентный каркас и среда разработки (при наличии соответствующих компонентов). Общая абстракция компонента-контейнера является основной, представленной каркасом [1, 3]; стандартный компонент Text, редактор составных текстовых документов, — её реализация. *Составной документ* — иерархия визуальных объектов, каждый их которых может быть контейнером. *Контейнер* может содержать сколько угодно других объектов, в добавок к его так называемому основному содержимому (текст — для компонента Text) [1, 1].

Абстракция каркаса *объект хранения* представляет сохраняемые в файл объекты и лежит в основе иерархии типов объектов из которых могут состоять составные документы Блэкбокс. Объекты хранения могут образовывать произвольные графы. «Граница» графа определена его *доменом* (domain). Это объект, который представляет всю совокупность объектов хранения, которые могут быть выгружены в файл или загружены из файла как единое целое. Составной документ Блэкбокс (в частности, текстовый) состоит из иерархии объектов изображения (визуальных объектов) и моделей[1], которые все суть объекты хранения, являющиеся частью одного домена. [2, Документация модуля Stores]

Сравнение составных текстовых документов Блэкбокс, которые (как контейнеры) могут содержать произвольные (внедрённые в текст) объекты, требует общего (не зависящего от конкретного типа объекта) метода сравнения для случаев, когда специальный, более точный механизм не доступен. Означенная в заголовке задача возникает, т. к. все внедрённые в текст объекты являются объектами хранения.

Решение и тестовая реализация. Очевидное решение напрашивается из сути объекта хранения — выгрузить сравниваемые объекты в файлы и сравнить полученные последовательности байт. Данное решение (в качестве общего) кажется вполне приемлемым, хотя и не может учесть временного (не сохраняемого) состояния объектов документа, поэтому возможны случаи счесть разные для пользователя объекты одинаковыми или наоборот.

[1] Подробнее см. [2, Путеводитель, Объекты изображения (вьюшки, views)]

Файл[2] в системе Блэкбокс также является абстракцией. Это делает решение практически применимым, т. к. позволяет использовать специальные реализации файлов, ориентированных на специфику задачи, не связанную с (постоянным) хранением данных.

Прямая реализация. Модуль PrivCS1 выгружает каждый объект хранения в отдельный файл и сравнивает содержимое файлов. Используется простейшая реализация файла в памяти PrivFixedMemFiles с минимально необходимым набором операций и фиксированной максимальной длинной:

```
TYPE File* = POINTER TO RECORD (Files.File)
    len-: INTEGER; data-: ARRAY 4096 OF BYTE
  END;
```

Процедура сравнения PrivCS1.EqStores:

```
PROCEDURE EqStores* (s1, s2: Stores.Store): BOOLEAN;
  VAR w: Stores.Writer; f1, f2: PrivFixedMemFiles.File;
BEGIN ASSERT(s1 # NIL, 20); ASSERT(s2 # NIL, 21);
  f1 := PrivFixedMemFiles.New();
  w.ConnectTo(f1); w.WriteStore(s1); w.ConnectTo(NIL);
  f2 := PrivFixedMemFiles.New();
  w.ConnectTo(f2); w.WriteStore(s2); w.ConnectTo(NIL);
  RETURN EqBytes(f1.data, f1.len, f2.data, f2.len)
END EqStores;
```

Оптимизация: последовательное сравнение при выгрузке второго объекта. Модуль PrivCS2 — использует две специализированные под задачу реализации файла. Первая сохраняет последовательность байт, вторая сравнивает поступающие данные с уже сохранёнными. Тестирование выявило некорректность метода в общем: абстракция файла допускает перезапись частей выгружаемой последовательности; в случаях перезаписи промежуточные данные второго объекта хранения не совпадают с окончательными первого.

Оптимизация: построение битовой карты равенства при выгрузке второго объекта. Модуль PrivCS3 — скорректированный вариант PrivCS2, учитывающий произвольность записи. Вторая реализация файла строит битовую карту равенства выгружаемой последовательности и данной. Оптимизация по памяти — $\frac{7}{8}FileSize(s2)$, по скорости (грубо) — сравнение байтовых массивов заменяется проверкой взведённости всех битов карты.

[2]Модуль Files компонентной библиотеки системы Блэкбокс, см. [2].

```
CONST maxLen = 4096;
TYPE
  File = POINTER TO RECORD (Files.File)
    len: INTEGER; data: ARRAY maxLen OF BYTE
  END;
  Writer = POINTER TO RECORD (Files.Writer)
    base: File; pos: INTEGER
  END;
  CmpFile = POINTER TO RECORD (Files.File)
    data: File; map: ARRAY maxLen DIV 32 OF SET; len: INTEGER
  END;
  CmpWriter = POINTER TO RECORD (Files.Writer)
    base: CmpFile; pos: INTEGER
  END;
...
PROCEDURE EqStores* (s1, s2: Stores.Store): BOOLEAN;
  VAR w: Stores.Writer; f1: File; f2: CmpFile;
    i, n, m: INTEGER; res: BOOLEAN;
BEGIN ASSERT(s1 # NIL, 20); ASSERT(s2 # NIL, 21);
  f1 := NewFile();
  w.ConnectTo(f1); w.WriteStore(s1); w.ConnectTo(NIL);
  f2 := NewCmpFile(f1);
  w.ConnectTo(f2); w.WriteStore(s2); w.ConnectTo(NIL);
  res := f1.len = f2.len;
  (* ... тестовая выдача разницы *)
  IF res THEN
    n := f1.len DIV 32; m := f1.len MOD 32;
    i := 0; WHILE res & (i < n) DO res := f2.map[i] = {0..31}; INC(i) END;
    res := res & ((m = 0) OR (f2.map[i] = {0..m - 1}))
  END;
  RETURN res
END EqStores;
```

Тестовый модуль. PrivTest — документ, содержащий исходник общего для всех методов сравнения тестового модуля и текст с набором тестов[3]. Модуль PrivTest экспортирует команду CmpStores активации произвольной процедуры сравнения двух объектов хранения, заданной по имени; команды задания параметров CmpStores (в частности, читающих их из близлежащего текста); и реализацию тестовых объектов хранения.

```
MODULE PrivTest;
  IMPORT Log, DevCommanders, Stores, Views, Meta, TextModels;
```

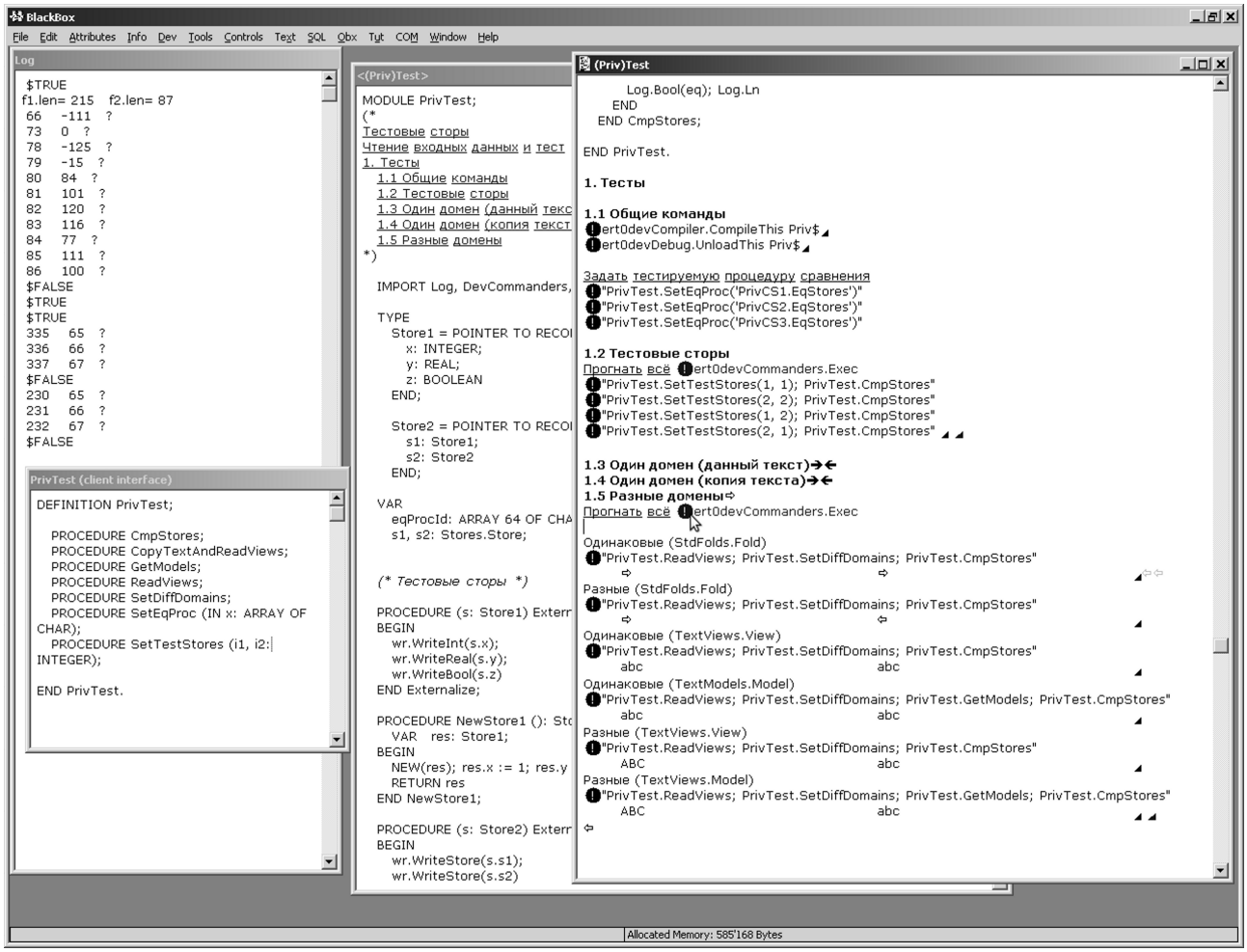
[3]Пример использования текста как интерфейса [1, 2.1].

```
  ...
  VAR
    eqProcId: ARRAY 64 OF CHAR; (* Параметры команды CmpStores: *)
    s1, s2: Stores.Store; (* устанавливаются другими командами *)
  ...
  PROCEDURE CopyTextAndReadViews*;
    VAR p: DevCommanders.Par; t: TextModels.Model;
      r: TextModels.Reader; v: Views.View;
  BEGIN ASSERT(DevCommanders.par # NIL, 20);
    p := DevCommanders.par; DevCommanders.par := NIL;
    t := TextModels.CloneOf(p.text); t.InsertCopy(0, p.text, p.beg, p.end);
    r := t.NewReader(NIL); r.ReadView(v); s1 := v; r.ReadView(v); s2 := v;
    ASSERT(s1 # NIL, 60); ASSERT(s2 # NIL, 61)
  END CopyTextAndReadViews;

  PROCEDURE SetDiffDomains*;
  BEGIN ASSERT(s1 # NIL, 20); ASSERT(s2 # NIL, 21);
    s1 := Stores.CopyOf(s1); Stores.InitDomain(s1);
    s2 := Stores.CopyOf(s2); Stores.InitDomain(s2);
    ASSERT(s1.Domain() # s2.Domain(), 60)
  END SetDiffDomains;
  ...
  PROCEDURE CmpStores*;
    VAR eq, ok: BOOLEAN; item: Meta.Item;
      val: RECORD (Meta.Value)
        x: PROCEDURE (s1, s2: Stores.Store): BOOLEAN
      END;
  BEGIN ASSERT(s1 # NIL, 20); ASSERT(s2 # NIL, 21);
    ASSERT(eqProcId # "", 22);
    Meta.LookupPath(eqProcId, item);
    ok := item.Valid() & (item.obj = Meta.procObj);
    IF ok THEN item.GetVal(val, ok) END;
    IF ~ok THEN
      Log.ParamMsg("Объект '^0' не существует
либо не является процедурой сравнения", eqProcId, "", "")
    ELSE
      eq := val.x(s1, s2); Log.Bool(eq); Log.Ln
    END
  END CmpStores;

END PrivTest.
```

Ниже приведён снимок экрана системы Блэкбокс, в которой открыт документ PrivTest (в двух окнах — начало текста с тестами и начало документа), журнал с результатами выполнения команды (около указателя мыши; процедура сравнения 'PrivCS3.EqStores'), и текст с интерфейсом модуля PrivTest.



Список литературы

[1] Темиргалеев Е. Э. Компонентное ПО и "текст как интерфейс". Примеры в Блэкбокс. **Вестник науки**. Сборник научных работ преподавателей, аспирантов и студентов физико-математического факультета ГОУ ВПО «ОГУ». Выпуск 10. — Орёл: Картуш, 2011. — 266 с. // URL: `http://oberoncore.ru/library/temir_comp-soft_text-ui_bb-ex`

[2] Русский перевод документации Блэкбокс. Блэкбокс — базовая сборка (дата обновления: 09.04.2011). URL: `http://www.inr.ac.ru/~info21/software.htm` (дата обращения: 17.04.2012)