CSE4312F12 Project Solution
ROI

Damien Gruel (cse23089@cse.yorku.ca)
Ludovic Lavalette (cse23088@cse.yorku.ca)

November 10, 2012

Note

- A customer elicitation session was held during class on Tuesday November 6, 2012. If you were not there sure to catch up with a fellow student who was there.
- This template is handed out *caveat emptor*. There may be errors and wrong information. It is ultimately your responsibility to elicit the correct requirements from the customer and to ensure that you satisfy the customer goals and specify correct output from the input.
- Your are required to correct any errors or ambiguities in this template and use this template to produce your final requirements document.

Revisions

Date	Revision	Description
10 October 2012	1.0	Initial customer elicitation
15 November 2012	2.0	Initial Student solution
1 December 2012	3.0	Final Student solution

Contents

List of Figures

List of Tables

1 Elicitation of customer goals

Our customers are the CEO and IT manager of Investment Corp. They desire an easy-to-use application to keep track of their return on their investments (ROI). A requirements elicitation session with our customers yielded the following issues and goals:

Each month, the customer receives a portfolio statement. The customer is not interested in keeping track of individual stocks, bonds etc. in their portfolio. What they would like to know is how the fund is doing thus far, i.e. over all data as well as over a specified evaluation period. There are free applications such as GnuCash for calculating ROI, but such applications require too much input information and are too complex for what is required. Customers also want to compare their return on their investment portfolios with respect to standard benchmarks [?, ?]. Benchmark data (when available) is either at year-end or year-to-date.

Customers might have multiple portfolios e.g. one for themselves and one for their spouse. All returns on investments are expressed in percentage per annun and all calculations must be done to industry standards. Customers might enter deposits into the investment account at arbitrary dates during the year. They may also withdraw money at arbitrary dates. This will affect the ROI calculation.

The input data for each portfolio is maintained by the customer as a CSV (comma separated value) text file as in Fig. ??. This allows them to keep track of their data on their smart phones or other devices. On the Windows desktop, double clicking on the file opens in Excel as in in the figure. The input also reports agent fees and, possibly a benchmark. For market values, cash flow (positive in, negative out) and agent fees, where no value is shown the default is zero.

Customers receive statements (sometimes monthly or every couple of months and always at the end of each year Dec. 31) from their investment brokers. The statement has a bottom line viz. the total value of all their investments to date (which includes bonds, stocks, etc.). Customers enter the total value of their portfolio at that date. For example, customers might receive a statement dated December 31, 2006 for $10,000. This is the value of the portfolio at the end of December 31, 2007, and is entered as $10,000 dated January 1, 2007. All entries thus reflect the value of the portfolio at the beginning of the day with deposits and withdrawals occurring during the rest of the day.

	A	B	C	D	E
2	**Description:** BMO RRSP, bonds and equities				
3	**Account#:** 478902				
4	**Email:** trudel@gmail.com				
5	**Address:** 4700 Keele Street, Toronto, M3J 1P3				
6	**Phone:** 416-736-2100 x70000				
7	**Evaluation Period:** 2008-01-01 to 2009-04-01				
8					
9	**Transaction Date**	**Market Value**	**Cash Flow**	**Agent Fees**	**Benchmark**
10	**2007-01-01**	**100000**			
11	2007-02-01	105000			
12	2007-02-11	105000	15,000		
13	2007-06-30	134000			
14	**2008-01-01**	**145000**			15.00%
15	2008-05-14	155000	16000		
16	2008-09-10	190000	-45000		
17	2008-09-30	172000			
18	**2009-01-01**	**230000**			35.00%
19	2009-02-20	350000	17000		
20	2009-04-01	390000			42.00%
21					

```
Description: BMO RRSP, bonds and equities,,,,,
Account#: 478902,,,,,
Email: trudel@gmail.com,,,,,
Address: 4700 Keele Street, Toronto, M3J 1P3,,,,,
Phone: 416-736-2100 x70000,,,,,
Evaluation Period: 2008-01-01 to 2009-04-01,,,,,
,,,,,
Transaction Date,Market Value,Cash Flow,Agent Fees,Benchmark,
2007-01-01,100000,,,,
2007-02-01,105000,,,,
2007-02-11,105000,15000,,,
2007-06-30,134000,,,,
2008-01-01,145000,,,15.00%,
2008-05-14,155000,16000,,,
2008-09-10,190000,-45000,,,
2008-09-30,172000,,,,
2009-01-01,230000,,,35.00%,
2009-02-20,350000,17000,,,
2009-04-01,390000,,,42.00%,
,,,,,,,,
```

Figure 1: Excel CSV input file

Customers do not want to pay a lot of money for the software and so they are prepared to forgo many things — a minimalistic product is expected. The product may be used via a command line interface (it may also have a simple GUI or can be mounted as a properly secured web application).

2 Context Diagram

Provide a context diagram with precise description of monitored and controlled variables. Indicate the entities in the environment. Note that in the sequel below we provide the precise nature of the input. You must elicit the precise outputs that are required.

3 Dictionary

The dictionary is incomplete

CSV: Comma Separated Value file format used to store tabular data in which numbers and text are stored in plain-text form that can be easily written and read in a text editor.

Customer: The user of the software system.

Evaluation Period: a start and end date (provided by the user) for the portfolio history over which the return on investment is calculated.

GIPS: Global Investment Performance Standards [1]

Investment broker: Runs the portfolio on behalf of the customer and supplies portfolio accounts.

Portfolio statement: List of all investments and current value.

Portfolio History: the historical data of investment performance over time that the customer stores about their investments as gleaned from their monthly or yearly investment accounts. Uusally stored by customers in a CSV file (see Figure 1).

TWR: Time Weighted Return (see [1])

4 E/R-descriptions

Fill this in ..

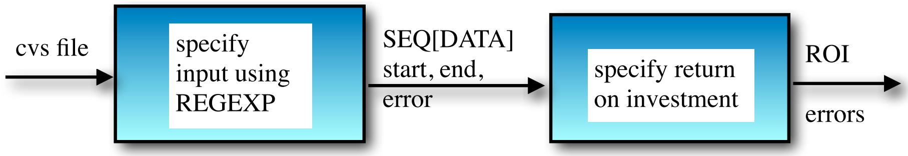
ENV1	Description	References

REQ2	Description	References

Note that you must calculate the compounded TWR (and its annualized value) for the complete period as well as for the evaluation period. The TWR is not always accurate. You must provide an accurate caculation (called precise-ROI).

5 Mathematical model

We provide below an incomplete mathematical model for ROI system. We define a valid input CSV file as a regular expression. Obviously there must be an error report for files that do not match the precise specification of input. We also provide the outline of a type to calculate the TWR and precise-ROI, which you must complete. You will also need a function table to ensure that all possible inputs (including faulty inputs) are handled.



type DATA

use DATE --see birthday book
carrier set $DATA \triangleq$
$Data(date : DATE, mv : \mathbb{F}, cf : \mathbb{F}, af : \mathbb{F}, bench : \mathbb{F} \cup \{Void\})$
require $valid\_date(date)$

--the above declaration induces an injective function with accessors
$Data : DATE \times \mathbb{F} \times \mathbb{F} \times \mathbb{F} \times \mathbb{F} \cup \{Void\} \rightarrowtail DATA$
const $mv, cf, af : DATA \to \mathbb{F}$
--market-value, cash-flow and agent-fees
const $bench : DATA \to \mathbb{F} \cup \{Void\}$ --benchmark
axiom $\forall d : DATA \bullet d = Data(date(d), mv(d), cf(d), bench(d))$
--e.g. $DATA = \{Data('2007\text{-}02\text{-}11', 105000, 15000, 0, Void), \cdots\}$

type input-csv

use type $REGEXP, DATA$
param $csvfile : \mathbb{S}$
--input parameter
query $tr : SEQ[DATA]$
--sequence of transactions
require $csvfile \in VALID\_FILE$
query $start, end : DATE$
require $csvfile \in VALID\_FILE$

--see Fig. ?? for the details

type roi

use $input(cvsfile)$
param $cvsfile : \mathbb{S}$
query $report : \mathbb{S}$

const $roi : \mathbb{F}$
const $roi\_twr : \mathbb{F}$
--you need to specify
--other outputs here

module control

--specification pattern for transfomational programs

event $roiReport \leftarrow invoke$
any $file : \mathbb{S}$
use $roi(file)$ **as** r
do $roiReport := r.report$

--dynamic use of module **roi**
--instantiate $cvsfile$ as $file$

Figure 2: Module specification of return on investment

type csv-input

use type $REGEXP, DATA, DATE$ --we let $\epsilon = \{\text{“”}\}$, eol $= \{\text{“\textbackslash n”}\}$ etc.
carrier set $DATA \triangleq Data(date : DATE, mv : \mathbb{F}, cf : \mathbb{F}, af : \mathbb{F}, bench : \mathbb{F} \cup \{Void\})$
param $csvfile : \mathbb{S}$ --input parameter
query $tr : SEQ[DATA]$ --sequence of transactions defined by axiom below
  require $csvfile \in VALID\_FILE$
query $start, end : DATE$
  require $csvfile \in VALID\_FILE$

const $VALID\_FILE : REGEXP$
 $\triangleq HEADER \cdot PARAMETERS \cdot \text{eol} \cdot ROW \cdot *(\text{eol} \cdot ROW) \cdot *(\text{“,”}|\text{eol})$
const $HEADER : REGEXP$
 $\triangleq *(HLINE \cdot eol)$
const $HLINE : REGEXP$
 $\triangleq *(\Sigma \backslash \text{eol}) \backslash (EV\_PER \cdot *\Sigma)$
const $PARAMETERS : REGEXP$
 $\triangleq EV\_PER \cdot DATE\_STR \cdot \text{“\_to\_”} \cdot DATE\_STR \cdot *\text{“,”} \cdot \text{eol} \cdot COL\_HEAD$
const $COL\_HEAD : REGEXP$
 $\triangleq +\text{“,”} \cdot eol \cdot$
  “Transaction_Date,Market_Value,Cash_Flow,Agent_Fees,Benchmark” $\cdot *\text{“,”}$
const $EV\_PER : REGEXP \triangleq$ “Evaluation_Period:_”
const $ROW : REGEXP$
 $\triangleq (DATE\_STR \cdot \text{“,”} \cdot FLOAT \cdot \text{“,”} \cdot (FLOAT|\epsilon) \cdot \text{“,”} \cdot (FLOAT|\epsilon) \cdot$
  $\text{“,”} \cdot (FLOAT \cdot \text{“\%”}|\epsilon) \cdot *\text{“,”})$
const $s2d : DATE\_STR \rightarrow DATE$ --see birthday book for $DATE$
const $s2f : FLOAT \rightarrow \mathbb{F}$ --deferred, $FLOAT$ is the string version of $\mathbb{F}$
const $f2s : \mathbb{F} \rightarrow FLOAT$ --deferred, see your favourite programming language
const $d2s : DATE \rightarrow DATE\_STR$ --deferred
const $s2optf[G] : (FLOAT|\epsilon) \times G \rightarrow \mathbb{F} \cup G$ --string-to-optional float
  where $\forall G \bullet s2optf \in (FLOAT|\epsilon) \times G \rightarrow \mathbb{F} \cup G$
  --parameter G is a set such as $\{Void\}$ or a default value such as $\{0\}$
const $f : ROW \rightarrow DATA$
dummy $w : ROW$ and $s_0, s_1, s_2, s_3 : \mathbb{S}$
axiom 1: --definition of function f that maps a row string to data
   $w \in (d2s(d) \cdot \text{“,”} \cdot s_0 \cdot \text{“,”} \cdot s_1 \cdot \text{“,”} \cdot s_2 \cdot \text{“,”} \cdot s_3 \cdot *\text{“,”})$
  $\wedge\ (s_4 \cdot \text{“\%”} = s_3 \vee s_4 = s_3 = \epsilon)$
 $\Rightarrow f(w) = Data(d, s2f(s_0), s2optf(s_1, 0), s2optf(s_2, 0), s2optf(s_4, \text{Void}))$
query $error : \mathbb{B} \triangleq textfile \notin VALID\_FILE$ --definition of $tr, start, end$
axiom 2: --definition of $tr, start, end$
   $csvfile \in VALID\_FILE \Rightarrow$
   $(\exists h, foot, s, e : \mathbb{S}; data : SEQ[ROW]$
    $|\ \ h \in HEADER \cdot EV\_PER \cdot s \cdot \text{“\_to\_”} \cdot e \cdot *\text{“,”} \cdot \text{eol} \cdot COL\_HEAD$
    $\wedge\, data \in SEQ[ROW]$
    $\wedge\, end \in *(\text{','}|eol)$
    $\wedge\, textfile \in h \cdot (\cdot i | 0 \le i < \#data \bullet eol \cdot data(i)) \cdot foot$
   $\bullet\ \ tr = (\cdot i | 0 \le i < \#data \bullet < f(data(i)) >$
    $\wedge (start = s2d(s)) \wedge (end = s2d(e))$
)

Figure 3: Type csv-input

6 Acceptance Tests

Very incomplete. You need a large number of tests including error tests

Test Case ID	T1
Description	Verify that return on investment (compounded TWR) is calculated correctly
Requirement IDs tested	R1?
Type	Positive
Initial State	A directory containing the CSV file in Figure ??
Action	Execute the ROI system on the CSV file
Consequences	The ROI system reports the compounded TWR as 144.08%

Test Case ID	T2
Description	Verify that return on investment (compounded TWR) is calculated correctly
Requirement IDs tested	R1?
Type	Positive
Initial State	A directory containing the CSV file in Figure ?? with the evaluation period from 2007-01-01 to 2009-04-01
Action	Execute the ROI system on the CSV file
Consequences	The ROI system reports the compounded TWR as 82.49%

7 Requirements Traceability matrix

Requirement ID	**Test Case IDs**
R1	T1, T2, ...
R2	...
R3	...

A REGEXP

A set of strings is used as the model for regular expressions. We use prefix operators for the Kleene closure (e.g. $*x$ where x is a regular expression such as {'hello'}) and iteration at least one or more (e.g. $+x$) rather than suffix operators. Note that where there is no confusion we use 'hello' instead of {'hello'} where the set is a singleton.

type REGEXP
carrier set *REGEXP* --set of all regular string expressions **axiom** $REGEXP \subseteq \mathbb{P}(\mathbb{S})$ **carrier set** $\Sigma \triangleq \{``0", ``1", ``2", \cdots, ``a", ``b",$ etc., all printing characters$\}$
dummy $x, y, z : REGEXP$ **dummy** $s, t, u : \mathbb{S}$ **axiom** $\forall s \in \Sigma \bullet \{s\} \in REGEXP$ **const** $0 : REGEXP \triangleq \{\}$ --zero is the unit element of alternation **const** $1 : REGEXP \triangleq \{``"\}$ --1 is the unit element of concatenation --we also use ϵ instead of 1 **const infix** "$\|$" : $REGEXP \times REGEXP \to REGEXP$ --alternation **const infix** "$\cdot$" : $REGEXP \times REGEXP \to REGEXP$ --concatenation **const prefix** "$*$" : $REGEXP \times REGEXP \to REGEXP$ --iteration zero or more times **const prefix** "$+$" : $REGEXP \times REGEXP \to REGEXP$ --iteration one or more times **axiom** $s \in x \vert y \equiv s \in x \vee s \in y$ **theorem** $x \vert 0 = 0 \vert x = x$ **axiom** $s \in x \cdot y \equiv (\exists t, u \vert s = t \cdot u \bullet t \in x \wedge u \in y)$ --note that $t \cdot u$ is concatenation over $SEQ[\mathbb{S}]$ **theorem** $1 \cdot x = x \cdot 1 = 1$ --1 is the identity of concatenation **const infix** "^": $REGEXP \times \mathbb{N} \to REGEXP$ --use this operator by raising the second argument like an exponent **axiom** $x^n = (\cdot i \,\vert\, 0 \leq i \leq n \bullet x)$ --concatenation quantifier --e.g. $x^3 = x \cdot x \cdot x$ **theorem** $x^0 = 1$ **axiom** $s \in *x \equiv (\exists n : \mathbb{N} \bullet s \in x^n)$ **axiom** $s \in +x \equiv (\exists n : \mathbb{N}_1 \bullet s \in x^n)$

Figure 4: Type REGEXP for regular expressions over printing characters

We may use type REGEXP to specify a *FLOAT_STRING* as follows.

Date	Market Valu	Cash flow	duration	wealth	TWR	TWR*100(%)
2011-01-01	10000		365.000			
2011-09-07	10500		116.000	1.050		
2012-01-01	11000			1.048		
					0.100	10.000

Date	Market Valu	Cash flow	duration	wealth	TWR	TWR*100(%)
2011-01-01	10000		365.000			
2011-09-07	10500	500		1.050		
2012-01-01	11000			1.000		
					0.050	5.000

Figure 5: TWR as a good approximation

$$\begin{aligned} FLOAT\_STRING = & \text{'+'}Inf && (1)\\ & |\text{'-'}Inf && (2)\\ & |NaN && (3)\\ & |(\text{'-'}|\text{'+'}|\epsilon)\cdot({}^{*}d\cdot\text{'.'}|\epsilon)\cdot{}^{*}d\cdot((\text{'e'}\cdot(\text{'-'}|\epsilon)\cdot^{+}d)\,|\,\epsilon) && (4)\\ d = & \text{'0'}|\text{'1'}|\cdots|\text{'9'} && (5) \end{aligned}$$

In the above we use the convention that 'e', for example, really stands for the single set {'e'}.

B Precise calculation of ROI

The TWR is only an approximation to the real time-weighted return, as in Fig. ?? (where one can see that the infusion of \$500 in cash reduces the ROI).

Fig. ?? shows where the approximation goes badly wrong. In this case, the investment advisor made a huge profit for our client and the gain is 16.19%, whereas the TWR shows the advisor as making a loss of 5.58%.

Clearly, a more precise method is called for.

B.1 Compound interest

Suppose you invest \$1000 for 5 years at 10% per annum. So we know that after the first year we have $1000 * R = 1000 * 1.10 = 1100$ (where R is the rate of return multiplier, i.e. $R = 1.1$). For 5 years we have $1000 * R^5 = 1610.51$. The general formula is

$$PV * (1 + r)^n = FV$$

A	B	C	D	E	F	G
Date	**Market Value**	**Cash flow**	**duration**	**wealth**	**TWR**	**TWR*100**
2011-01-01	10		365			
2011-09-07	9	10000	116	0.9		
2012-01-01	10500			1.04905585		
					-0.0558497	-5.5849735
				which indicates a loss!		

Figure 6: TWR as a bad approximation

where r is the interest rate as a decimal (e.g. 0.1, i.e. 10%), and n is the number of periods. If $R = 1 + r$ then we have $PV * R^n = FV$

For the second example in Fig. **??**, it is not so simple as we are not adding amounts at regular intervals (cash flow in/out is irregular). We may use n as a day (i.e. 1/365 of a year) and we get:

$$FV = (PV * R^{365/365}) + (500 * R^{116/365})$$

i.e. we weight the initial money ($PV = 10000$) by the full year (365 days) of daily return. The input of cash in Septmember (which is 116 days to the end of the years) is weighted in that proportion. The equation van be re-written

$$(PV * R^{365/365}) + (500 * R^{116/365}) - FV = 0$$

and we can solve for the root of the polynomial (e.g. using Newton-Raphson) to obtain R. This gives us a net gain of 4.923%. The TWR calculated it as 5%, which is only approximately correct.

The general formulas is

$$tr(1).mv * R^{\Delta days(i)} + \tag{6}$$
$$(\Sigma i : \mathbb{N} | 1 < i < n \bullet tr(i).cf * R^{\Delta days(i)}) - tr(n).mv = 0$$
$$\textbf{where } n = \mathrm{card}(tr)$$
$$end = tr(n).date \tag{7}$$
$$\Delta days(i) = (end - tr(i).date)/365$$