RuleBurst®
prevent • detect • cure

White Paper

Time-based Reasoning:

Dealing with complex changes in rules, rates and circumstance

June 2007

Contents

1 Introduction

Many public and private sector organisations work in a complex environment, where calculations are required to determine the entitlements and obligations of customers against an ever-changing base of policy rules. This is particularly true of major Government agencies administering tax, social security/social services, immigration and compensation. It is also required for private institutions such as banks and insurers.

This paper outlines RuleBurst's sophisticated solution to dealing with change in a complex policy environment. RuleBurst's elegant approach reduces risk for implementation, is faster to implement, has lower TCO for both development and ongoing maintenance; and improves comprehension for customers and end-users. The solution caters for the fact that 3 critical elements of the calculation of a customer's entitlements and obligations are subject to constant change:

1. **The rules that apply to a customer.** For example, a new benefit component may be legislated by Government for social security payments; or a special rule for high risk debtors may be introduced by a bank.
2. **The reference data that is used to calculate specific rates of payment.** For example, every quarter a Government agency may update its Tax Credit thresholds and payments. An insurance company may review its risk thresholds monthly.
3. **The data (evidence) relevant to a customer.** For example, a care provider receiving a Government subsidy may provide services to a large number of end-customers who are constantly moving in and out of care. Similarly, the well-being of the customers of a health insurer may fluctuate from month to month.

Where each of these elements is subject to constant change, the stakeholders in an IT system require at minimum the following:

- A simple way of representing data for a period of time, over which a calculation can then be made (for example, over a year);
- A simple way of showing the results of these calculations, identifying the rates applied to each time period, and aggregating these into a total amount;
- The capacity to readily change the rules and reference data and measure the impact of that change on the customers affected by the rule changes;
- Confidence in the performance of the system, knowing throughput and scalability are as good as or better than alternative approaches

Previous solutions to this problem have involved a less than optimal approach, involving the slicing of time periods with a great deal of data input/output required – particularly in the case of complex calculations relying on a significant number of input factors.

RuleBurst has developed a world-leading approach that vastly simplifies this calculation process, allowing host applications to take advantage of new capabilities in RuleBurst 9. The business benefits of this approach are:

- It is significantly faster (at least 3 times) than conventional coding, both for initial rule definition and deployment, and ongoing maintenance;
- The expression of the logic is more concise and comprehensible than using application code, in particular for business analysts as opposed to programmers;
- Retrospective calculations can be explained in a transparent way to end users and customers, and the explanation stored for review and appeal;
- The resulting calculation engine can be integrated and used by any case management system including for self service.

2 The Business Problem

The problem is best illustrated by a representative example, using a simplified fictional tax credit. Note that virtually identical issues arise in a number of business contexts where rules and rates change over time, for example:

- Calculation of premiums payable by companies under a health insurance policy.
- Payment of Pensions to aged people.
- Calculation of interest rates to debtors/creditors of a bank.
- Entitlement calculations for social security benefits.
- Calculation of tax payable by citizens or businesses.

2.1 Tax Credit Example

Take a simple tax credit, payable to a family based on the following factors:

- Whether or not the family has a sole parent or a couple (ie whether or not the claimant of the tax credit is single or in a recognised relationship);
- Whether the claimant and/or their partner are working;
- The income of the claimant and/or their partner;
- For each child in the family:
    - The age of the child;
    - Whether the child is studying.

For this tax credit:

- The income thresholds and amounts payable are reviewed quarterly in line with CPI changes;
- On 1 May 2008, a new set of rules applies to increase the amount of the tax credit for families where 1 parent works and the other does not;
- By default, payments are made fortnightly by the Tax Office, although a person can also claim at the end of a tax year for all the amounts within that tax year;
- The tax year runs from 1 January to 31 December.

Given this background, at least 2 use-cases are common:

1. **Retrospective claim.** A couple have their first child, and within 6 months of the child's birth make a claim for payment. In that period of time, the female member of the couple has returned to work after a period of maternity leave.
2. **Retrospective change of circumstance.** A couple is receiving ongoing fortnightly payment for 2 children, and the eldest child leaves school. They do not notify the tax agency until some months after the child has left school.

2.2 Use case 1: retrospective claim

Key input data and dates:

Date	Relevant Change	Type of Change
1 Jan 2008	Quarterly rate change	Rate
1 Feb 2008	Child born	Circumstance
1 April 2008	Quarterly rate change	Rate
16 April 2008	Female member of the couple returns to work	Circumstance

1 May 2008	New rules effective for families with 1 parent working	Rules
1 July 2008	Quarterly rate change	Rate
10 July 2008	Claim made by the family for the tax credit	Circumstance

Key outputs:

- The rate applicable between each change date;
- The total amount of tax credit due to the family for the period from the birth of the child to the date of claim;
- The ongoing rate of tax credit due to the family assuming no further changes in circumstance;
- The likely total amount payable to the family to the end of the tax year.

2.3 Use case 2: retrospective change of circumstance

Key input data and dates:

Date	**Relevant Change**	**Type of Change**
1 Jan 2008	Quarterly rate change	Rate
1 April 2008	Quarterly rate change	Rate
1 May 2008	New rules effective for families with 1 parent working	Rules
1 July 2008	Quarterly rate change	Rate
3 August 2008	Eldest child leaves school	Circumstance
1 October 2008	Quarterly rate change	Rate
1 December 2008	Family notifies tax office that child left school	Circumstance

Key outputs:

- The rate that should have applied from the 3rd August;
- The difference between the total amount that should have been paid to the family between 3 August and 1 December, and the amount actually paid to them;
- The ongoing rate of tax credit due to the family assuming no further changes in circumstance;
- The likely total amount payable to the family to the end of the tax year.

3 The solution

3.1 Existing Market Solutions

Traditionally, the solution provided by most application software vendors to this problem has involved creating 2 separate rule-sets that interact only at defined points:

1. A set of rules to accurately calculate the specific rate/premium applicable at a given date (a point in time); and
2. A separate set of rules to aggregate the individual point in time calculations and where applicable sum them to a specific amount.

This approach shall be referred to as the "time slicing" approach, and it suffers from the following limitations:

- The total cost of ownership of building and maintaining the rules in conceptually separate sets (even if they are in the same physical code-base) is at least 3 times higher than in RuleBurst's latest approach outlined below.
- Separating out point in time calculations from those that apply over a period of time can lead to very complex data structures, which can be inefficient for processing in a high volume environment (for example tax or social security calculations where millions of customer records need to be processed in a timely manner).
- It can be difficult to cater for special rules such as **when** a new rate will apply from, where the point in time rules are separate to the aggregation rules.
- The separation of the different rule sets can make debugging difficult, particularly for cases involving lots of changes in circumstance or rate.
- Applying more rules such as "grandfathering" can be very complex (where an old rate applicable to a person can continue if they applied under an old set of rules which has subsequently changed).
- Typically, the aggregation rules are much harder to change in this solution pattern than the point in time rules, limiting the policy options available to the business.

The time slicing approach is very common in mainframe legacy environments, as well as packaged solutions from ERP or similar providers. It has also been used by RuleBurst in the past.

3.2 RuleBurst's Solution

RuleBurst now has a powerful solution that allows complex rule-sets to be "time aware", and operate simultaneously for specific points in time as well as across time periods. Available in RuleBurst Version 9, with a patent pending, this provides a revolutionary way of quickly and simply capturing and applying the rules required to accurately calculate changing rates.

As the worked example below illustrates, this approach involves incorporating the notion of time as a fundamental concept in the underlying rules engine.

This enables the following:

- An inherent property of every data item in the rule set (both inputs and outputs), is it's value at a given point in time;
- Inbuilt **temporal operators** are provided that tap into this property, including functions to calculate time-dependent items like:
    - Whether a particular condition is true for a given number of days/months/years in a specific time period;
    - The total amount for a data item based on complex logic spanning any given time period. For example, the total amount of a social security benefit over any given time period.
    - Whether or not a condition is true on, before or after a specified time period.

These functions enable logic which is natural to a person to be captured in a readily understandable way, very naturally handling conditions like the following:

- "You must not have more than three alcoholic drinks in any two hour period."
- "Retirement age is 55 if you started work before 1950, and 65 if you started work in or after 1950."
- "You are eligible for disability pension if you have been off work due to an injury for three consecutive months in any twelve month period."
- "Until the end of the current financial year, the levy is 1%, at which time it goes up to 2%. However, if your age at the end of the financial year is over 65, it will initially stay at 1%, and increase by 0.25% a year until it reaches 2%."

4 Worked Example

RuleBurst's solution is best illustrated using a worked example.

4.1 Pension Calculation Rules

For this worked example, a pension payment is payable based on the following rules:

- To receive their payment, the person must satisfy an age threshold:
  - Up until 1 January 2007, this age threshold was 55 years of age;
  - From 1 January 2007 inclusive, the age threshold is changed to 65 years of age.
- The standard daily rate of a person's benefit is calculated according to the following:
  - $5 per day regardless of marital status up until 1 January 2007;
  - After 1 January 2007, it is either:
    - $6 per day if the person is not married; or
    - $7 per day if the person is married.
- The actual daily rate paid to a person (the amount they are entitled to) is based on the following, regardless of which time period they fall into:
  - 1 x the standard daily rate if the person is not married;
  - 1.5 x the standard daily rate if the person is married.

4.2 The representation of these rules in RuleBurst 9

The rules below are actual "application code" from RuleBurst 9, capturing the business logic described above as rules in natural language.

4.2.1 Total Entitlement

the person's total entitlement for pension for the period = TemporalDailySum(the person's daily entitlement for pension, the start of the period, the end of the period)

4.2.2 Daily Entitlement

the person's daily entitlement for pension	
the standard daily rate of benefit	the person is not married and the person satisfies the age requirement
the standard daily rate of benefit * 1.5	the person is married and the person satisfies the age requirement
0	**otherwise**

4.2.3 Age Requirements

the person satisfies the age requirement if

both

TemporalBefore(2007-01-01) and
the person's age in years >= 55

or
both

TemporalOnOrAfter(2007-01-01) and
the person's age in years >= 65

4.2.4 Person's Age

the person's age in years = TemporalYearsSince(the person's date of birth)

4.2.5 Standard Daily Rate

the standard daily rate of benefit	
5	TemporalBefore(2006-07-01)
6	TemporalOnOrAfter(2006-07-01) and the person is not married
7	TemporalOnOrAfter(2006-07-01) and the person is married
uncertain	**otherwise**

4.3 The Scenario

Take a simple scenario, in which the person who will receive the pension:

- Is born on the 1st January 1950; and
- Was initially single, then married on 1 April 2007.

The assessment period is 1 January 2005 until 1 January 2020.

4.3.1 Input Timeline

The person is assessed over a period from 1 January 2005 until 1 January 2020, generating the following timeline for the inputs:

Date	Relevant Change	Type of Change
1 January 1950	The person is born	Circumstance
1 July 2006	Rate change for single/married people	Rate
1 January 2007	New rules for age criteria	Rules
1 April 2007	The person is married	Circumstance

4.3.2 Output Timeline

The inputs above generate the following results for the person. Note there is no change in result on 1 April 2007, as the person's rate does not change on that date (they do not satisfy the age requirements).

Date	Conclusion
1 January 2005	The person turns 55 The person's daily rate is $5 per day
1 July 2006	The person's daily rate is $6 per day
1 January 2007	The person's daily rate is $0 per day as they no longer satisfy the age requirements which have changed
1 January 2015	The person's daily rate is $10.50 per day

4.3.3 Decision report

The person's total pension entitlement from 1 January 2005 to 1 January 2020 is $23017.50.

The person's daily entitlement for pension is

- $0.00
- $5.00 from 1 Jan 2005
- $6.00 from 1 Jul 2006
- $0.00 from 1 Jan 2007
- $10.50 from 1 Jan 2015

Whether the person is married is

- False
- True from 1 Apr 2007

Whether the person satisfies the age requirement is

- False
- True from 1 Jan 2005
- False from 1 Jan 2007
- True from 1 Jan 2015

The person's age in years is

- 55 from 1 Jan 2005
- 65 from 1 Jan 2015

The person's date of birth is 1 Jan 1950

5 Conclusion

RuleBurst has more than 19 years of working in the area of complex time based rules, and has numerous solutions deployed worldwide that implement solutions to the problem of rules and data that change over time.

This experience, and our ongoing R&D investment has led us to an elegant solution to handle the 3 intersecting areas of change:

- Complex changes in policy and rules;
- Changes in rates and other reference data;
- Changes in customer circumstance.

This solution provides a clear and understandable way to handle cases that need calculation over a period of time, in which all 3 of the above inputs are changing constantly. It is also more efficient from a processing perspective, requiring less CPU time to perform calculations that stretch over a period. The net result for a customer is much lower TCO for building and maintaining time-based rules, and much higher accuracy and consistency in the calculation outputs. For organisations who often pay out billions in payments, this makes such a solution a mandatory requirement.

Customers and partners are already demanding the massive increase in productivity from use-cases that exercise the newer time-based reasoning capability in RuleBurst 9. We expect this takeup to increase as organisations modernise their legacy systems, and move to an architecture where time-based complexity is encapsulated in a comprehensible set of rule documents. Once encapsulated in this way, then packaged solutions integrated with RuleBurst become massively more capable of dealing with the complexity of Government regulation and internal policy.

