Contents

1 UPL: Universal Processing Language

1.1 History

Pocessing systems are using manually crafted application relays to handle card processing business rules. Being defined by business analysts these rules are fallen down to engineering teams informally. Approach we provide pushes card processing to solid background in a form of domain specific language common for all card plans analytics departments.

Having compact language we can formally build various translators for particular customers and existing processing systems. At the same time we provide reference back-end Erlang system implementation for transactions processing. Also DSL gives us a natural and easy verification strategies and compactifications.

This language could be easily extended to other domain aread like internet payment processing, shopping mall bonus programs, mobile operators tariff plans.

1.2 Objectives

The aim is to create small and compact language for payment transaction processing. Underlying instrumentation code should be KVS layer for storing transaction chains but naturally should be extended to different backends like Java, PL/SQL and other languages currently involved in banking card processing. We have several criterias to satistfy:

English self-explanatory domain language

Clean lexical tokens set

Optimized language forms for compact language core

Verification

Fasten Time-to-market

Taxonomy driven managability for billing plugins

1.3 Cards

Cards are tariff programs, set of rules that we plug to transaction processing. It feels like set of filters triggered each time we fire money movements on account with a given card defenition.

Listing 1: BNF

```
Program = card Name Currency Forms

Form = limit Amount
     | grace Amount days
     | credit CreditRules
     | rate ChargeRule
     | version Amount
     | deposit DepositRules
     | accounts AccountList
```

Example:

Listing 2: credit.card

```
card PLA_DEB USD
limit 20000
version 1.0
credit monthly 10%
```

1.4 Language Forms

Top level tariffs of billing rules are pluggable slangs that share some common part of the languages. These common part we will call language forms.

Listing 3: BNF

```
Direction = charge
          | withdraw

ChargeRule = Fixed + Percent
             of <amount | debt | credit | deposit | rate>
             limit <min Amount> | <max Amount>
             name Name
             to account Name

Periodically = monthly Amount
             | monthly Months -> ChargeRule
             | daily ChargeRule
             | annual ChargeRule

Account := <credit | rate | deposit> Name
```

1.5 Deposit

1.6 Credit