TATA CONSULTANCY SERVICES
Experience certainty.

A Model-Driven Approach for Specifying and Configuring Variability in Business Applications

VARY workshop of MoDELS 2011

Souvik Barat, Suman Roychoudhury and Vinay Kulkarni*
Tata Consultancy Services, India
{souvik.barat, suman.roychoudhury,vinay.vkulkarni}@tcs.com

Why Configurable Business Application

Indian Customer | US Customer | International Customer

Configuration Selection | Customer Login

Customer Login>>Us Customer Details

Basic Details

Customer SSN Number | Name | DOB

Local Address Details

Address Line1 | Address Line2 | City

State | ZIP Code

Permanent Address Details

Address Line1 | Address Line2 | City

State | ZIP Code

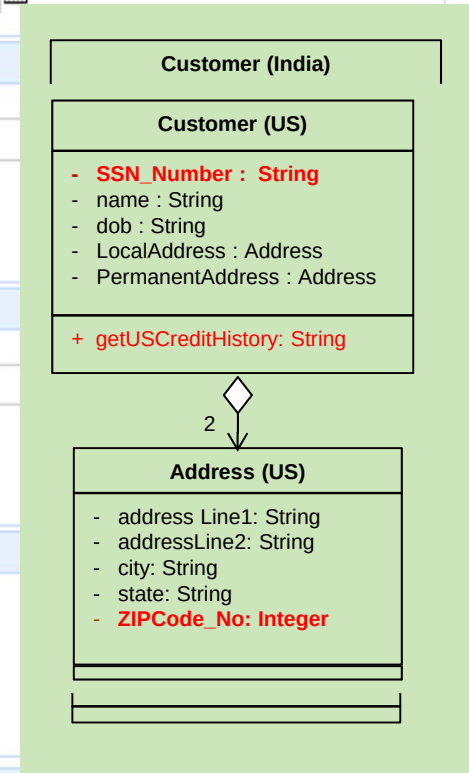
Credit History Details

Credit History Information:

```
Customer credit history for Us Configuration: Credit
worthy customer !
```

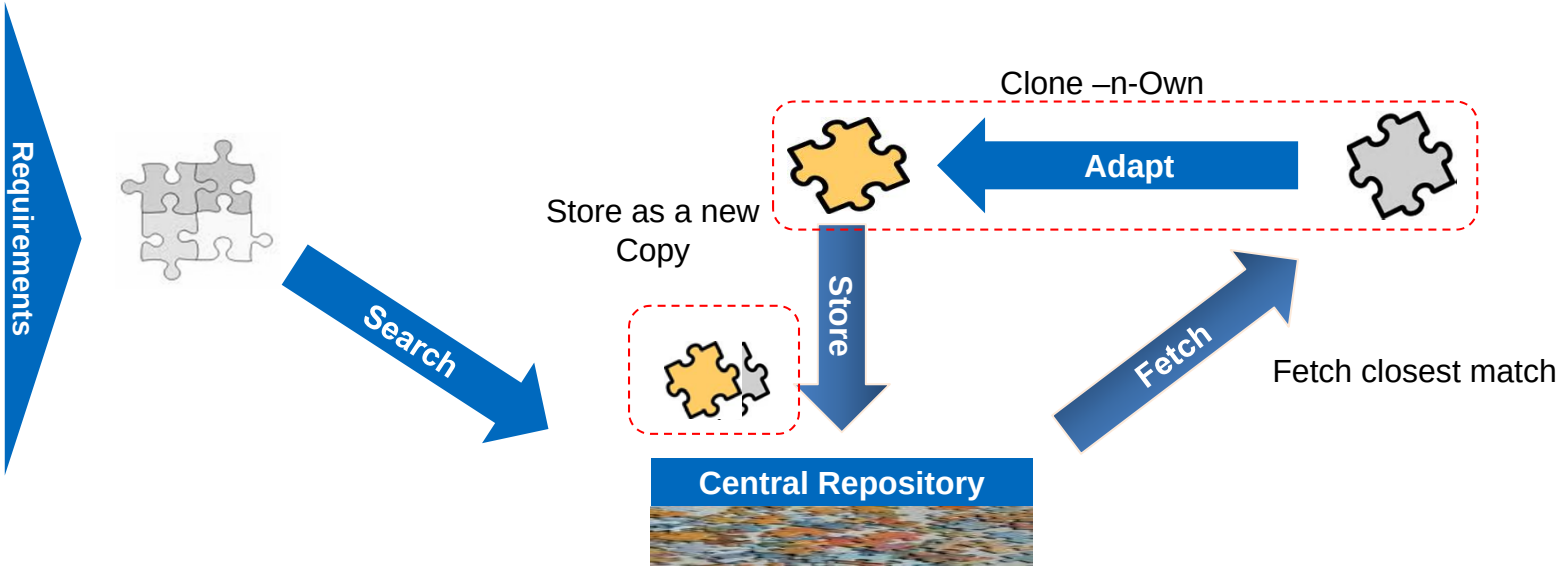
Get Credit History

Save



No two applications are same though they are for same business intent

Existing Approach in Practice



Approach for deriving new application/component with similar requirements

Issues with clone-n-own

- Leads to maintenance overhead
- Same change needs to be effected at multiple places
- Evolution traceability is lost

Use the concept of Product-Line

Challenges of using Product-line concept

- **Specifying Solution Space Variability**
  - **(Current State-of-art) Extend base Metamodel to capture variability, e.g. UML Stereotype**
- **Specifying Problem Space Variability**
  - **Specify problem space variability using Feature Model, etc but no mechanism exists to establish the relationship between problem space variability and solution space variability**
- **Context specific resolution technique**
  - **Automated resolution techniques with fixed semantics**

Our Approach

Implement desired variability (*Variability Realization Model)*

- Identify places where things differ i.e. *Variation Point*
- Identify *Variations* for each Variation Point
  - Not a one-step activity, introduce new variation on demand
- Specify constraints over Variations for internal consistency

Illustration

Specify *semantic interpretation* of identified Variation Points

- Use existing *Variation Point Type* (existing semantic)
- Define new Variation Point Type (define new semantic using QVT)

Illustration

Specify variability in closer-to-problem intuitive manner (*Variability Specification Model)*

- Describe variability in abstract manner, i.e. using *Feature model*
- Specify constraints over the available *Features*
- Define a set of internally consistent features leading to set of well-defined *Configurations*

Illustration

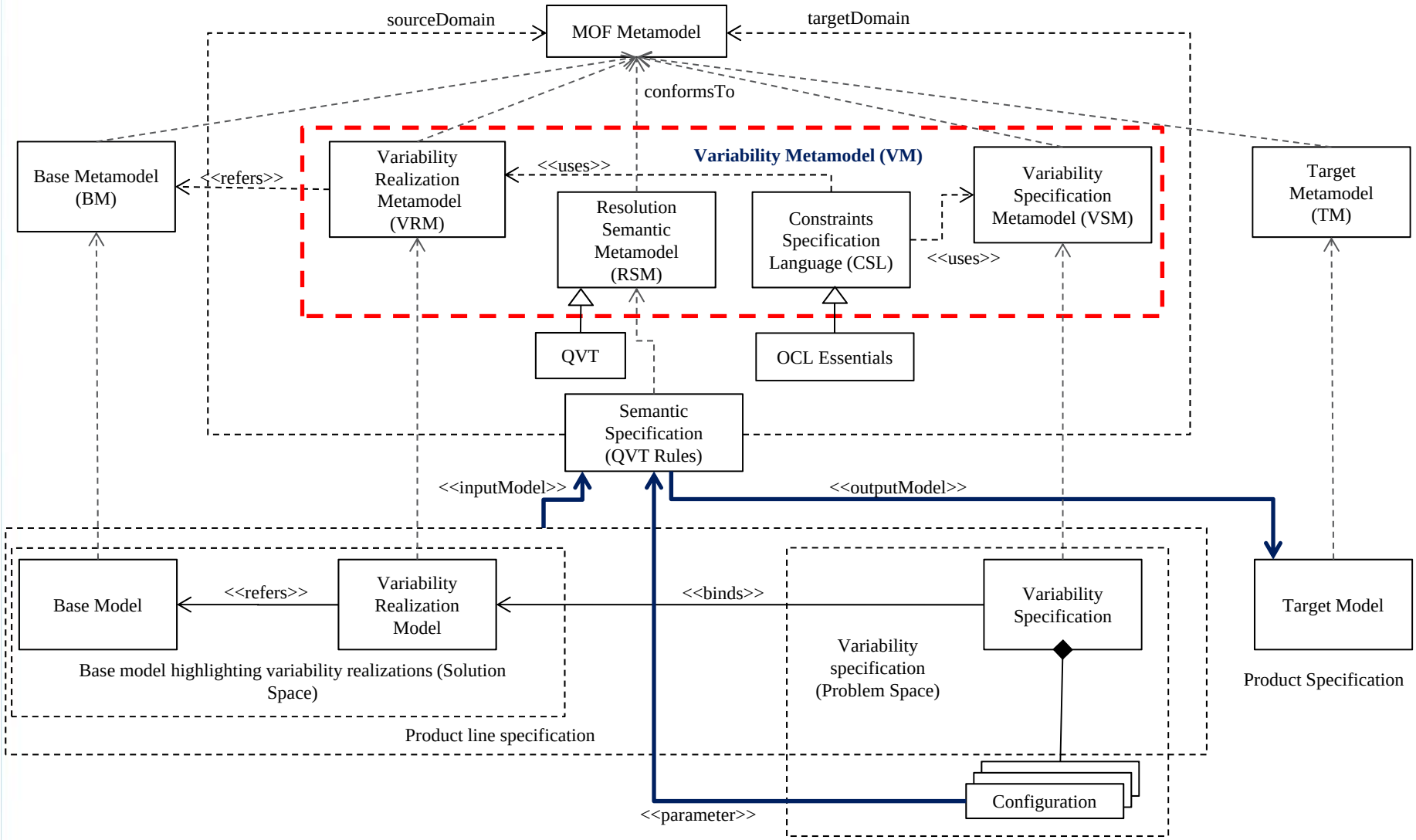
Bind Resolution and Variability models

Illustration

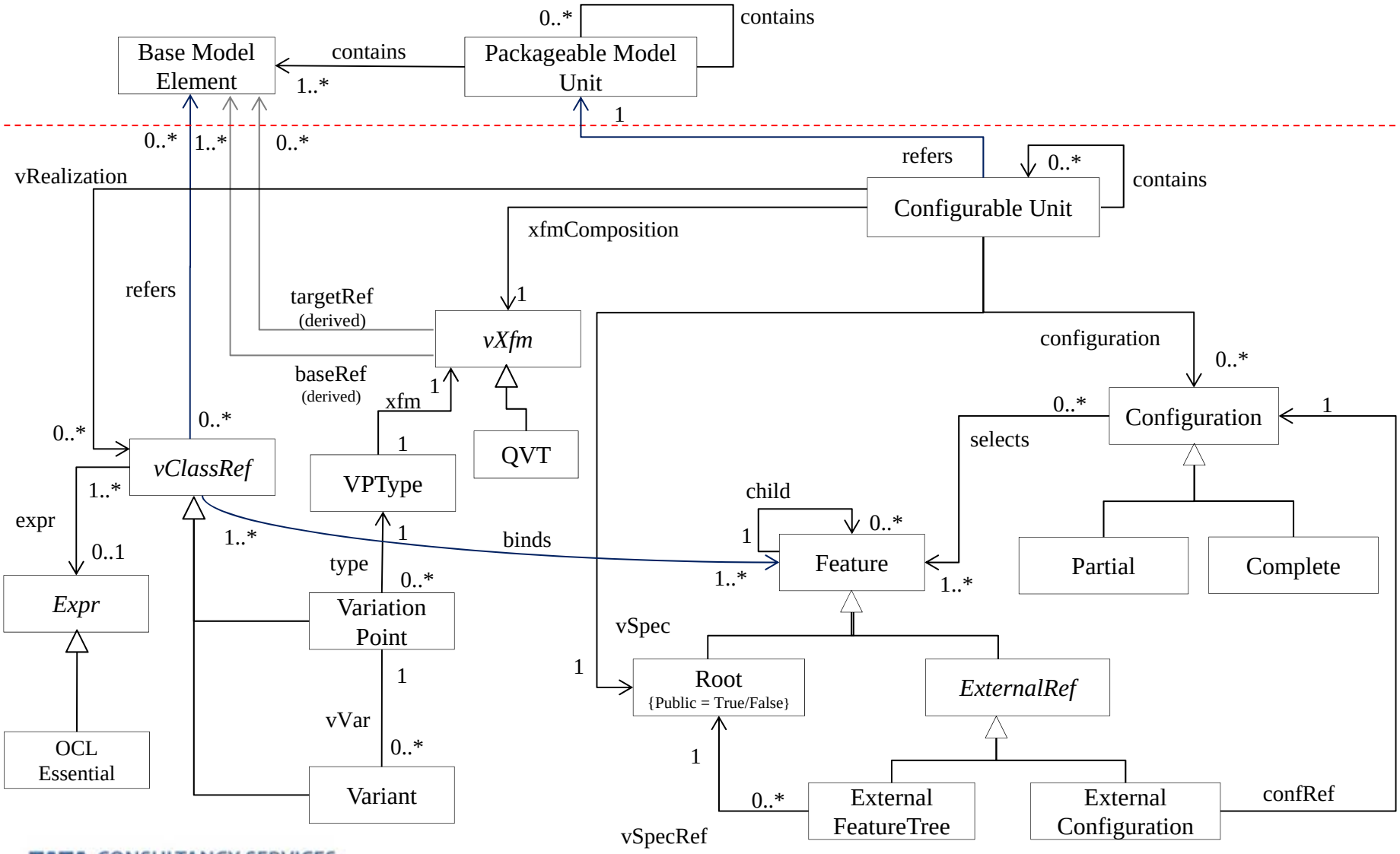
Resolve Variability

Illustration

Schematic Representation of Our Approach



Variability Metamodel



Conclusion

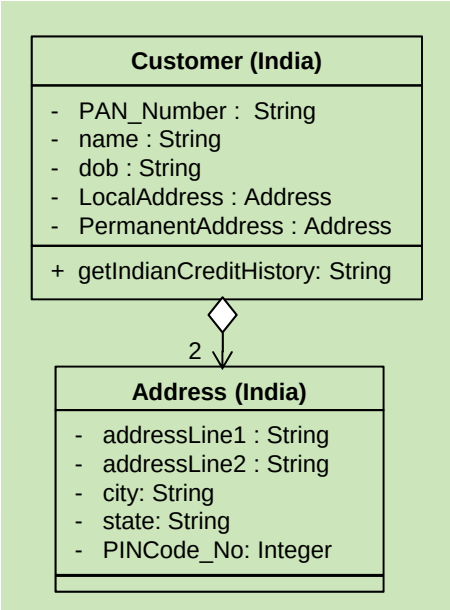
- **Variability of business application specifications are captured without modifying any base meta models, where a business application**
  - Comprises of Components, Service, Screen and Business process.
  - Component and Service specifications use UML Class Model
  - Screen uses MOF compliant purpose-specific metamodel
  - Business Process uses BPMN Metamodel

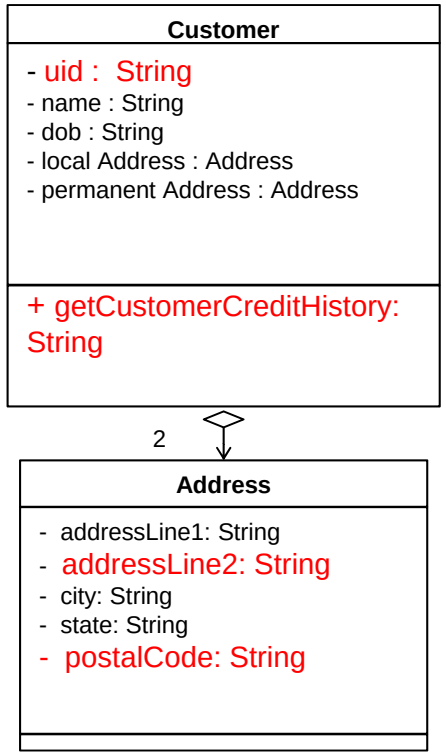
- **Problem space variability are specified in an abstract manner without consulting variability realization model.**

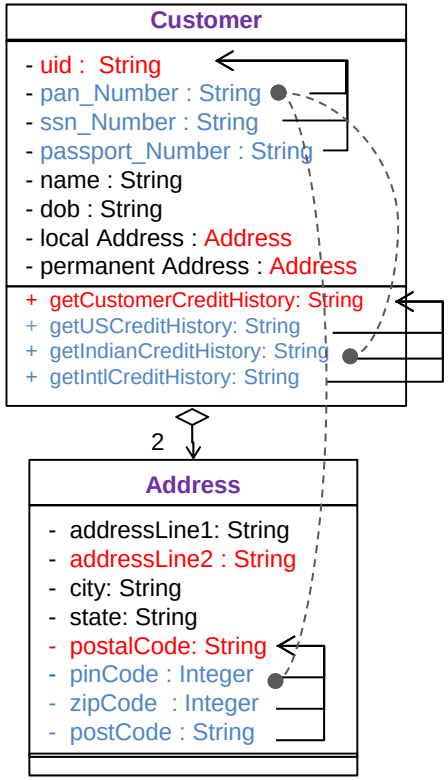
- **Model-to-model transformation is used to resolve variations.**

Questions

Implementing Variability – Variability Realization Model







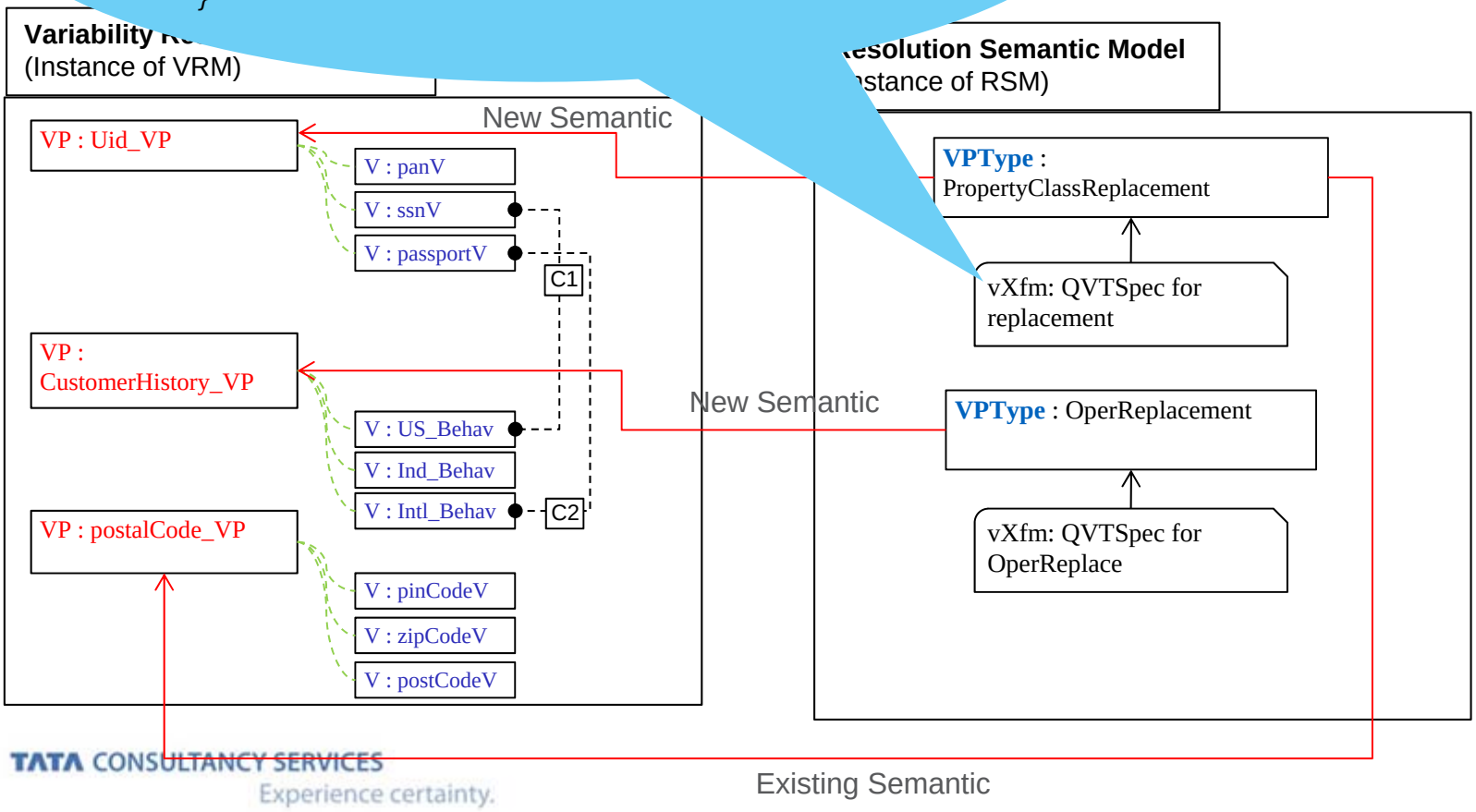
Consider any base version for defining Variability → Identify the places where things differ with the context and mark them as Variation Points → Specify Identified Variations for Variation Points → Specify Implementation Constraints

Key Concepts – Variation Point and Variation

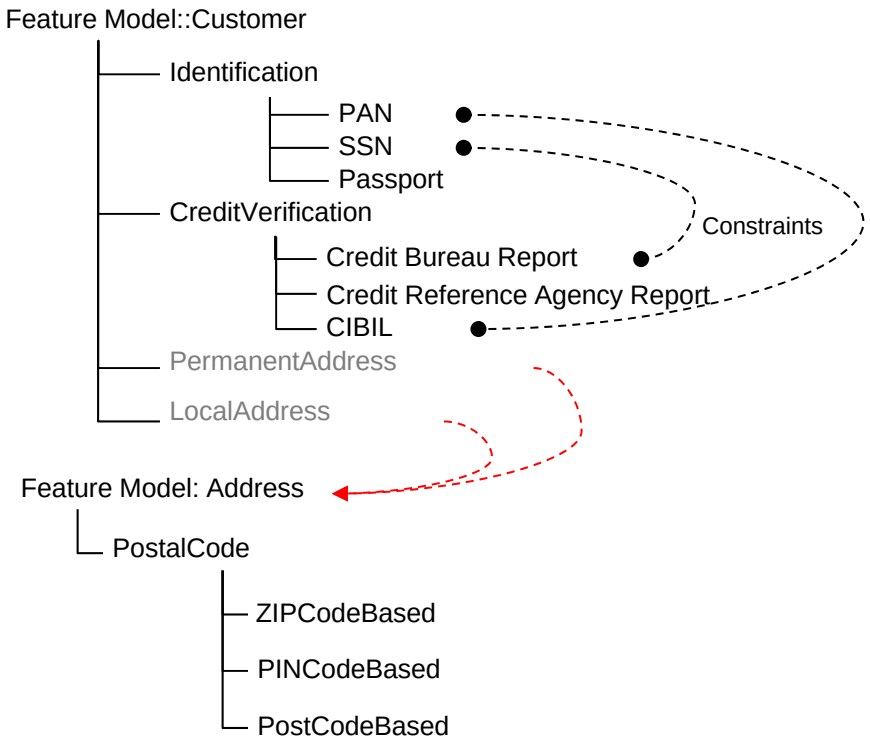
Specifying Semantic Interpretation

Semantic Specification

```
VPType: PropertyClassReplacement
Mapping CVL::VariationPoint:pVariationPointToProperty(in
      confName: String ) : TargetDomain::Class
when { self.ifVariationHasReferenceObject()}
{
    //QVT code for replacing variation Point by selected Variants for
      confName
}
```



Specifying Variability - Resolution Model



Indian Customer

1. Customer
1.1 Identification-> PAN
1.2 CreditVerification->CIBIL
1.3 PermanentAddress->PostalCode->PINCodeBased
1.4 LocalAddress->PostalCode->PINCodeBased

US Customer

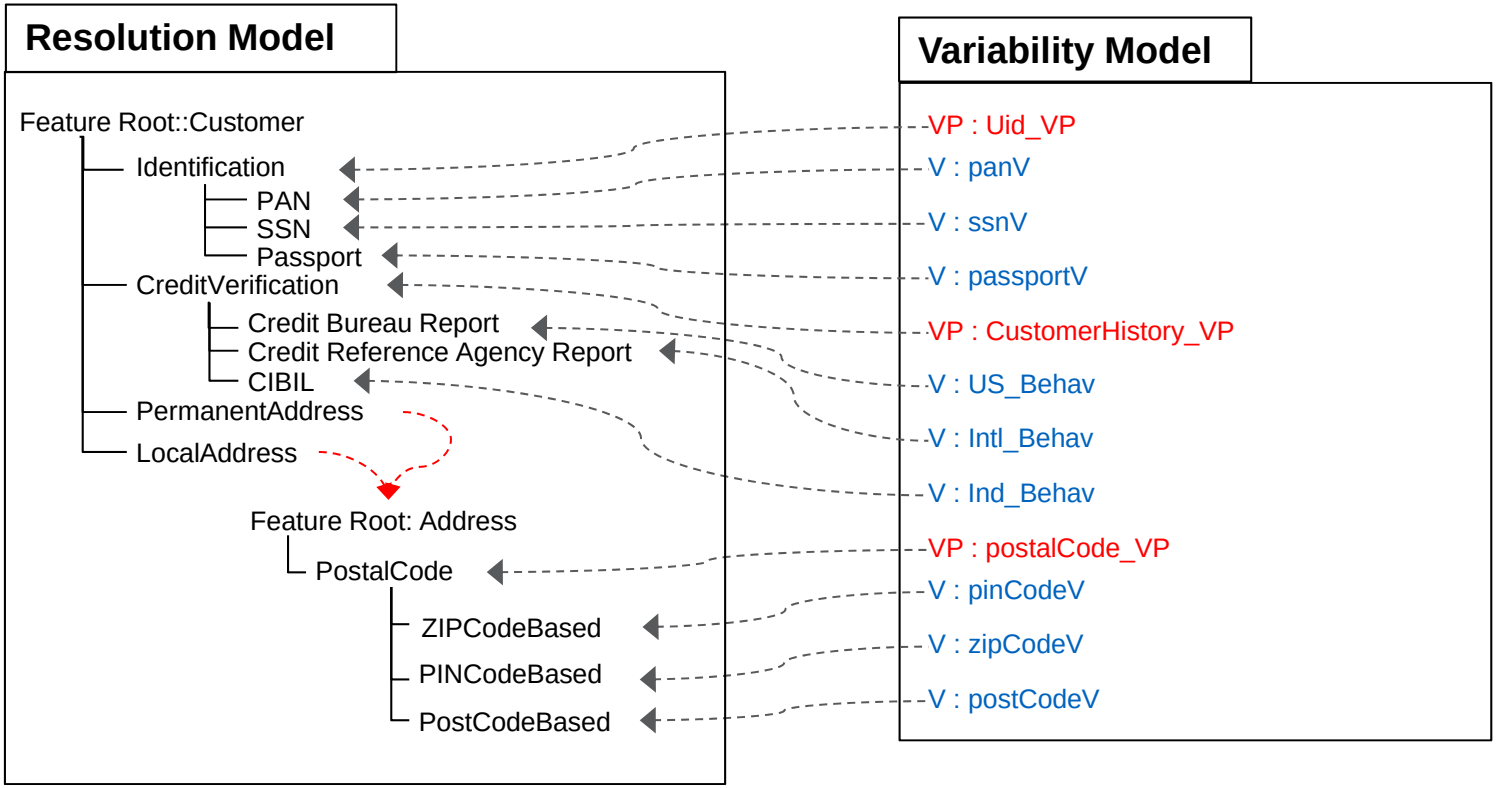
1. Customer
1.1 Identification-> SSN
1.2 CreditVerification->Credit Bureau Report
1.3 PermanentAddress->PostalCode->ZIPCodeBased
1.4 LocalAddress->PostalCode->ZIPCodeBased

International Customer

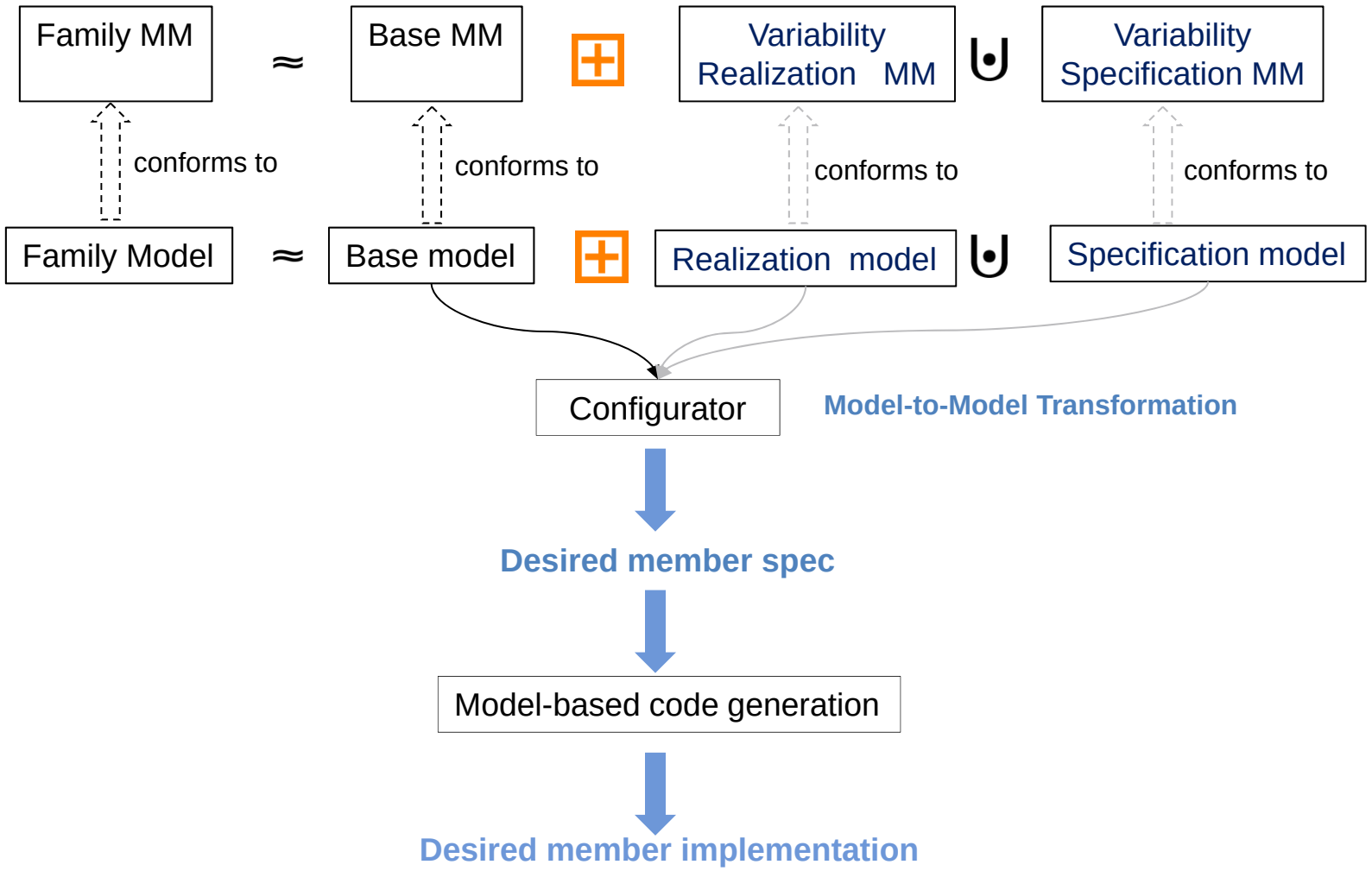
1. Customer
1.1 Identification-> Passport
1.2 CreditVerification->Credit Reference Agency Report
1.3 PermanentAddress->PostalCode->PINCodeBased
1.4 LocalAddress->PostalCode->PostCodeBased

Specify variability using feature model → **Specify constraints** → **Define Configurations**

Binding Resolution model and Variability model



Resolution


Family MM
≈
Base MM
Variability Realization MM
Variability Specification MM
conforms to
conforms to
conforms to
conforms to
Family Model
≈
Base model
Realization model
Specification model
Configurator
Model-to-Model Transformation
Desired member spec
Model-based code generation
Desired member implementation