



# An Ontology-based Method and Tool for Cross Domain Requirements Visualization

Nirav Ajmeri, Kumar Vidhani, Manoj Bhat, Smita Ghaisas

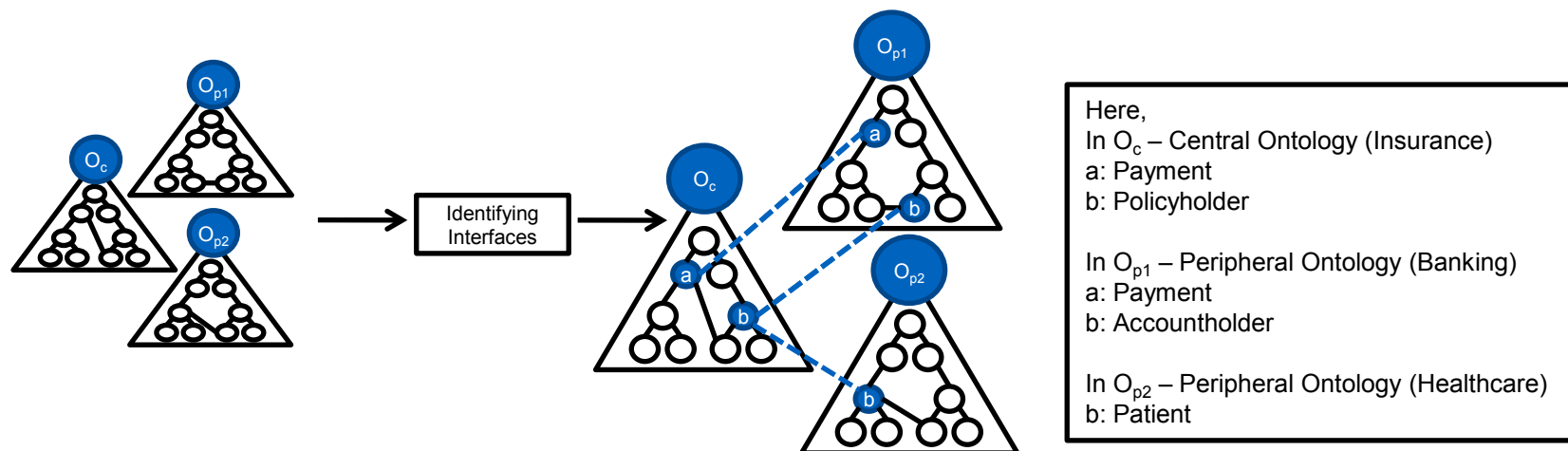
# Introduction & Motivation

- Complexity associated with cross domain scope of requirements is a challenge.
- Lack in understanding of interaction between domains may results in -
  - Incomplete requirements
  - Wrong Estimation
  - Schedule slippage
- Ontologies becoming standard way of representing domain knowledge.

Ontology-based method identifies *interfaces between multiple domains* and generate *conceptual model of requirements*

# Method For Identifying Interfaces

- **Semantically similar concepts** between multiple domain ontologies defined as ***interfaces***.
- Semantic Similarity is a combination of -
  - Syntactic Similarity based on the **syntactic structure** of concept
  - Sense Similarity based on **similar usage sense** of concept
  - Context Similarity based on context defined by **neighborhood** of concept



US\_1 - As an Insurance Company, we want *health insurance policyholder* to make *payment* for premium online through internet banking

# Method For Deriving Conceptual Model

## Criteria

1. Identify if interface concept represent functionality?
  - check if interface concept part of verb phrase or prefixed by a verb phrase.
2. Domain of the User Story?
  - map extracted concepts of User Story with Domain Ontology.

- If User Story executes in ***Peripheral domain***.

$$\text{Conceptual Model} = C_{\text{interface}} + C_{\text{complementary\_central}} + C_{\text{ack}} + \text{Associated Constraints} \#$$

- If User Story executes in ***Central domain***.

$$\text{Conceptual Model} = C_{\text{interface}} + C_{\text{complementary\_central}} + C_{\text{complementary\_peripheral}} + \text{Associated Constraints} \#$$

#  $C_{\text{interface}}$  – interface concept,  $C_{\text{complementary\_central}}$  – complementary concepts from central domain,  
 $C_{\text{complementary\_peripheral}}$  – complementary concepts from peripheral domain,  $C_{\text{ack}}$  – acknowledgement concept

# Experimental Results & Conclusion

Table1. Insurance domain User Stories found to be interacting with Banking and Healthcare domain

User Story	Description	User Story Domain	Interface and Complementary Concepts#
US_1	As Insurance Company, we want policyholders to make payment of premium online through internet banking.	Banking	<b>Payment</b> (C <sub>interface</sub> ), Premium, Risk, <b>Transaction</b> (C <sub>ack</sub> )
US_2	As a Claim Scrutinizer, I should be able to view insured's medical history in claims scrutiny screen.	Insurance	<b>Insured-Patient</b> (C <sub>interface</sub> ), Policy, Personal Details, Contact Details, Medical Report, Prescription, Disease, Healthcare provider
US_3	As Claims Manager, I want system to find out symptoms which are pointers to terminal diseases in health records furnished by customer.	Insurance	<b>Symptom</b> (C <sub>interface</sub> ), Disease, Insurance, Insurance policy, Syndrome, Organ

# C<sub>interface</sub> – interface concept, C<sub>ack</sub> – acknowledgement concept

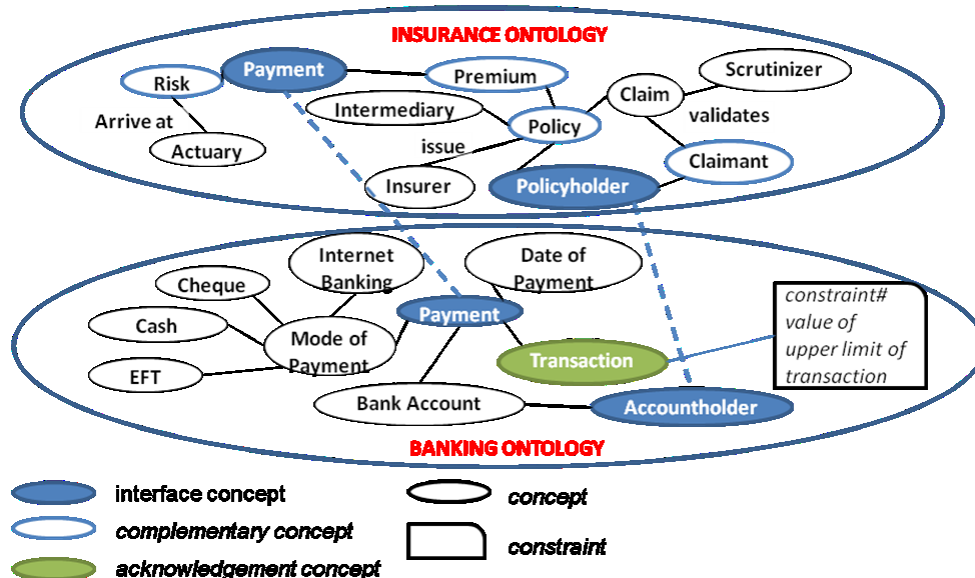


Figure 1. Conceptual model of US\_1

- Explicitly visualize cross domain scope of requirements.
- Completeness of ontologies is a precursor.
- A step towards improving completeness of requirements.



# Thank you!

email: {nirav.ajmeri, kumar.vidhani, manoj.bhat, smita.ghaisas}@tcs.com