Specification of Non-Functional Requirements: A Hybrid Approach

Unnati S. Shah <unnati.shah25@gmail.com>

PhD Candidate, Computer Engineering Department, S.V.National Institute of Technology, Surat, India

Co-Authors: Dr. Devesh C. Jinwala and Dr.Sankita J. Patel

Outline

- Background
  - Non-Functional Requirements
  - Ambiguity Problem
  - Impact of ambiguity resolution on completeness of requirements specification
- Motivation
- Proposed System Architecture
- ATM- A Case Study
- Conclusions

Types of Requirements

– Functional Requirements

- What the software will do?
- Description of the functions.
- Details of data to be held in the system

– Non-Functional Requirements

- How the software will do?
- Constraints on the functions.
- Quality, that the software must have

- In literature[12-15] we identify various definitions for NFRs.

Examples

The system must handle 1,000 transactions per second.
- Workloads
- Response Time

The system must have less than 1hr downtime per three months. • Availability

Functional Requirement

The system shall authenticate users before allowing access to customer data.

Non - Functional Requirement

a. The system shall use 256 bit RSA encryption for authentication handshake.

b. The system shall perform authentication within 3 seconds.

c. The system shall not allow login from mobile web browsers.

Examples

Unstated Non-Functional Requirements

The system shall provide a means to edit discharge instructions for a particular patient.

This sentence does not explicitly state a security requirement but implies security requirements for

1. Confidentiality (of patient's discharge instructions),
2. Integrity (when editing)
3. Accountability (who performed the edits).

Motivation

- As per our literature their exists numerous approaches/tools to specify NFRs.

1. NFR Framework based Approach[24-29]
2. UML model based Approach[24, 33-35]
3. Natural Language based Approach[37, 38]
4. Ontology based Approach[40-49]

- No Quality Models
Difficult to integrate with UML

- Lack of detailed quality attributes

- No information regarding application domain

- Domain knowledge support
- Unambiguous Representation
- Formal Specification

Ambiguity Problem!!!!!

"I like eating a lot!"

- "Eating is something I like a lot"?

It's ambiguous!

Or

- "Eating a lot is something I like"?

"I know a little Italian."

- "I know a small Italian person"?

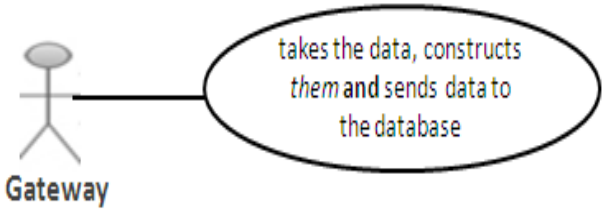
It's ambiguous!

Or

- "I understand a little of the Italian language"?

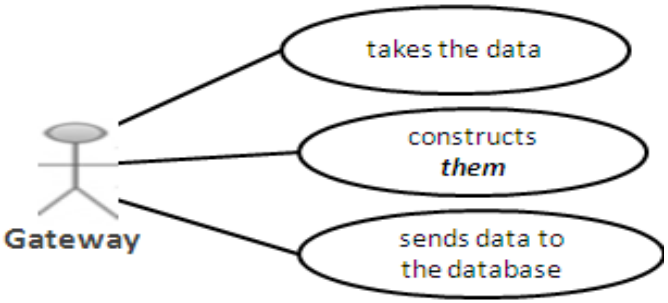
Impact of Ambiguity Resolution on Completeness of Requirements Specification

"The gateway takes the data, constructs them and sends data to the database".



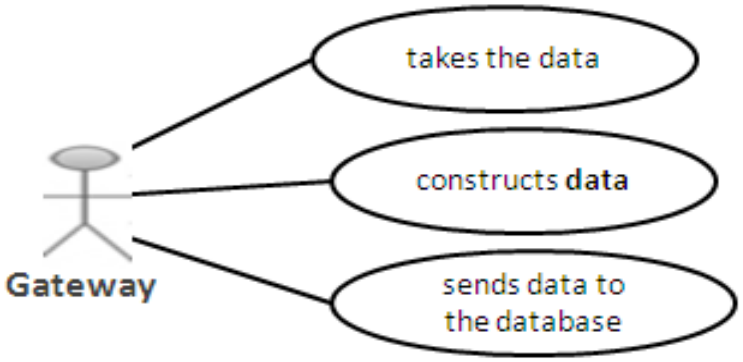
Use-case Diagram before Normalizing Requirements

Normalization



Use-Case Diagram after Normalization

Ambiguity Resolution



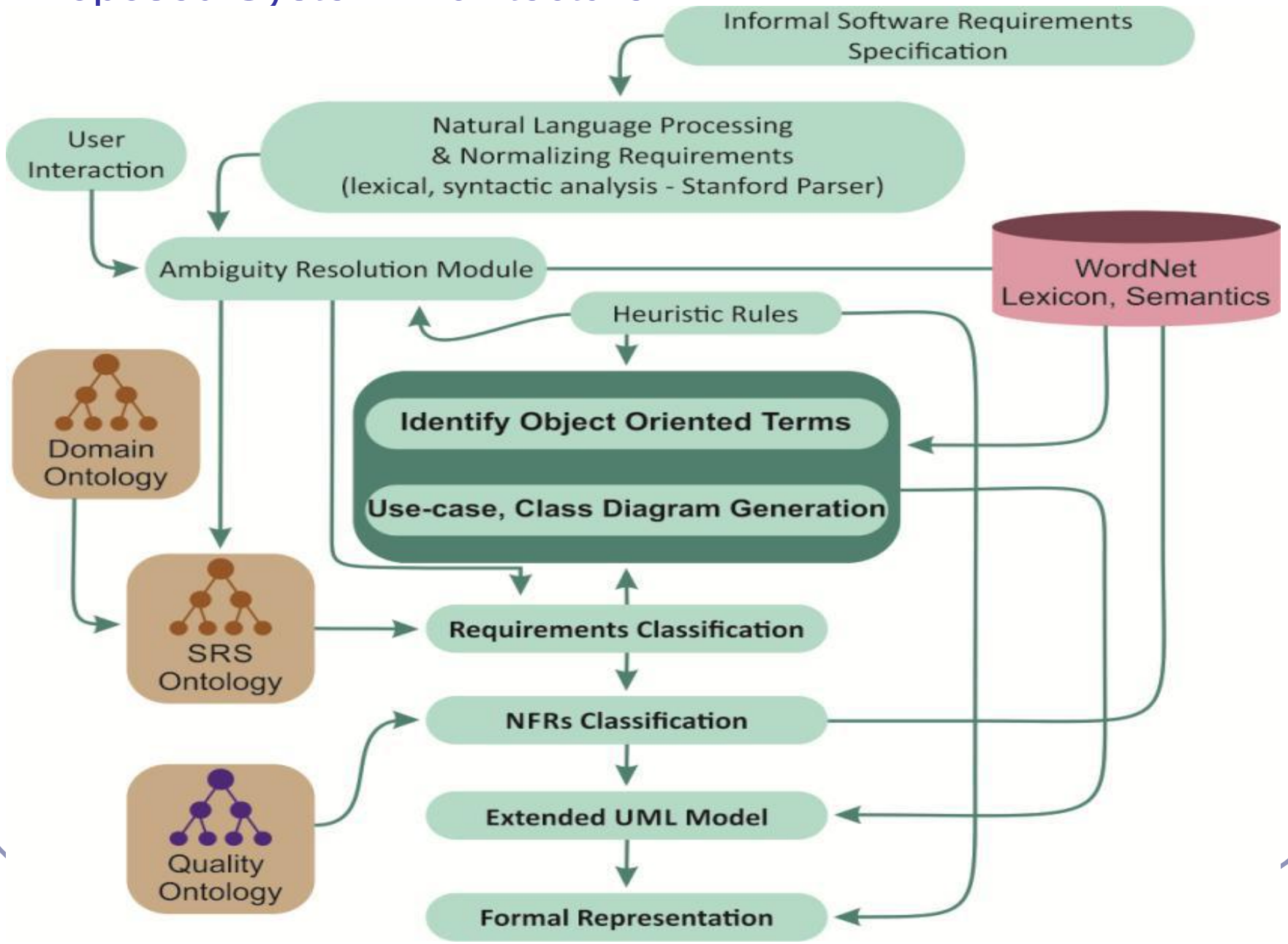
Use-case Diagram after Normalizing Requirements and Resolving Ambiguity

Problem Statement

- Our proposed approach mainly focuses on
  - Extracting possible NFRs from Requirements Documents
  - Provides specification after Resolving Ambiguity
- To achieve this, we use
  - A hybrid approach

    – combination of NLP, machine learning and ontology

Proposed System Architecture



Module	Task	Technology/ Approach	Output
Preprocessing	Sentence Splitting, POS tagging, Normalizing		Normalized Requirements
Ambiguity Resolving	Anaphora ambiguity (viz. event, verb, personal pronoun, possessive pronouns, Wh-pronoun, Wh-adverb), Coordination ambiguity and Attachment ambiguity	-Stanford Parse -WordNet -NLP -Heuristic Rules	Unambiguous Requirements
Create SRS Ontology	Based on prebuild domain ontology create a software requirements specification ontology	-Rule based Approach -Protégé tool	SRS Ontology
Create UML Models	-Identify OOT -Extract Relationship -Extract subject, object and dependency agent to identify possible actors -Extract verb phrases to identify possible use-cases	-Rule based Approach -Stanford Dependencies -WordNet	Use-case Model
Requirements Classification	-Stemming words -Extract nouns -Apply distance/similarity measures -Apply clustering algorithm	-Machine learning Approach -Hierarchical Clustering -Hamming Distance	Requirements Clusters
NFRs Classification	-Extract NFRs -Classify NFRs	-Ontology based Approach -Keyword search -Rule based Approach	Extracted NFRs
Extended UML Models	-Integrating NFRs in UML models	-Rule based approach	Extanded Use-case Model

ATM - A Case Study

Initial requirements

R1: The ATM interacts with the customer to gather transaction information.

R2: The bank computer gets the transaction information from the ATM to verify an account **and** to process a transaction.

R3: Each bank may be processing transactions from several ATMs at the same time.

R4: The customer interacts with the ATM network via the ATM.

R5: **It** must be very easy for **them** to use the ATM.

R6: The ATM network has to be available 24 hours a day.

R7: The ATM network should provide maximal security.

After Normalizing and Resolving Ambiguity

R1: The ATM interacts with the customer to gather transaction_information.

R2.1: The bank_computer gets the transaction_information from the ATM to verify an account.

R2.2: The bank_computer gets the transaction_information from the ATM to process a transaction.

R3: Each bank may be processing transactions from several ATMs at the same time.

R4: The customer interacts with the ATM_network via the ATM.

R5: It must be very easy for customer to use the ATM.

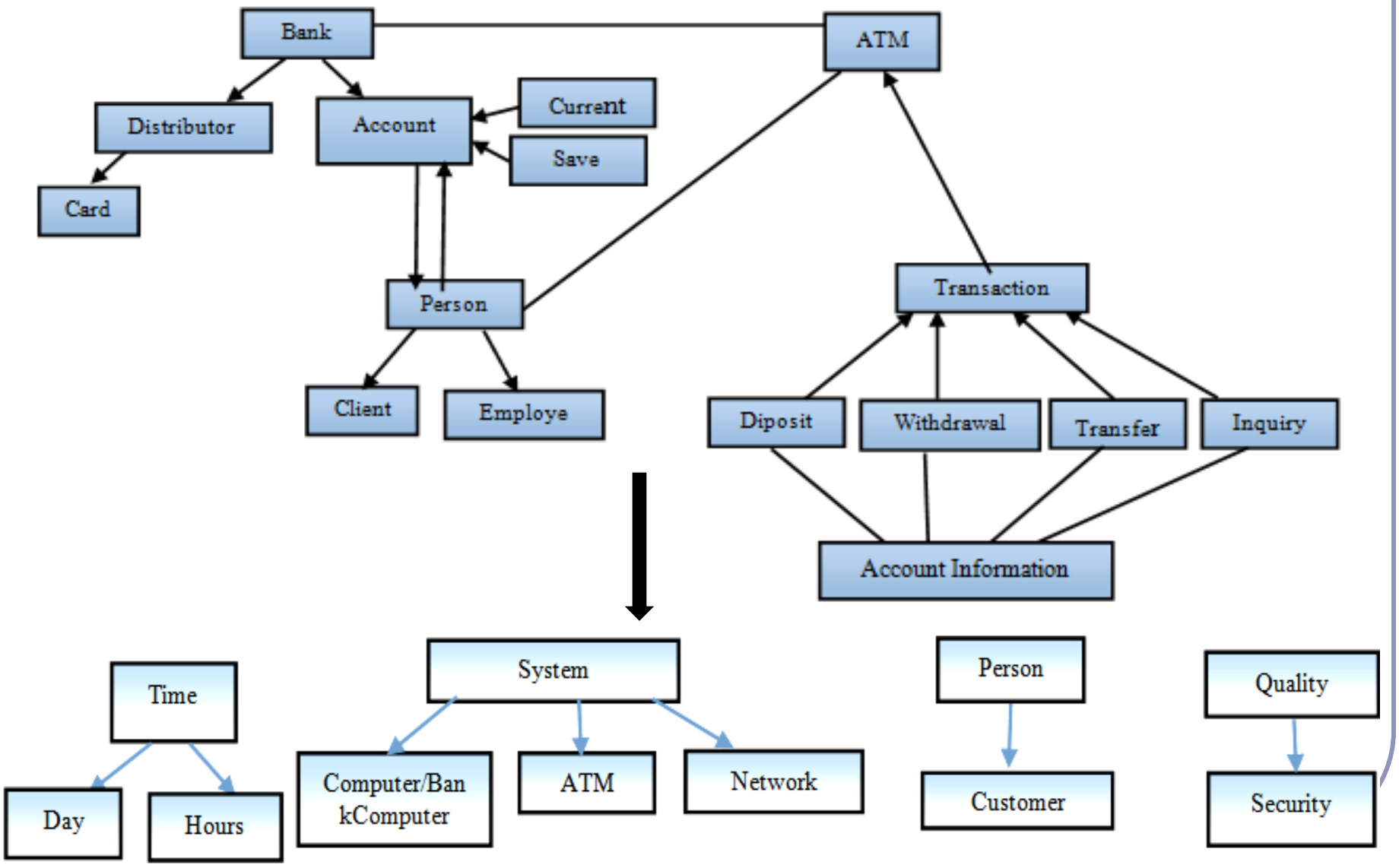
R6: The ATM_network has to be available 24 hours a day.

R7: The ATM_network should provide maximal security.

Extracted Objects

Noun	**Verb**	**Adverb/Vauge**
ATM	Interact	Several ATMs
Customer	gets	Same time
Transaction_Information	Gather	Very easy for user
Bank_Computer	Verify	Available 24 hours
Account	Process	Maximal security
Transaction	use	
Bank	Provide	
Time		
ATM_network		
Hours		
Day		
Security		

Create SRS Ontology using Pre-build



UML diagrams Generation

Procedure:

1. Extract Classes, Attributes and Methods:

Object Oriented Terms	Stanford Type Dependency
Subject /Class	Csubjpass (clausal passive subject) Nsubj (nominal subject) Nsubjpass (passive nominal subject) Xsubj (controlling subject)
Object/Class	Dobj (direct object) Iobj (indirect object) Pobj (object of a preposition)
Actor	Agent (agent)
Cardinality	Predet (predeterminer)
Attribute (String/number)	Acomp Advmod (adverbial modiffier) Amod (adjectival modifier) Num (numeric modifier) npadvmod: noun phrase as adverbial modifier
Methods	aux: auxiliary auxpass: passive auxiliary complm: complementizer rcmod: relative clause modifier xcomp: open clausal complement

Mapping from Stanford Type dependency to object oriented terms

UML diagram Generation (Cont…)

- Eliminate semantically same noun
  - Initial classes/ nouns

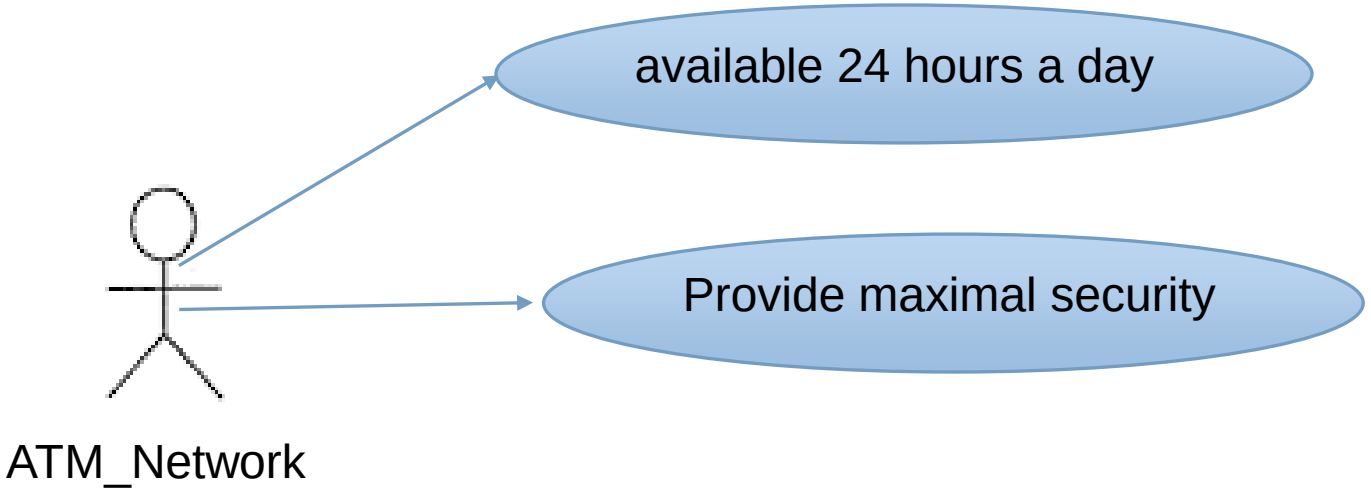
Noun/Class	Meaning from WordNet
Client	Customer,guest,node
Raam	No meaning available
Customer	Client
System	Arrangement, organization, scheme, system of rules
Security	Certificate, protection
Systems	Arrangement**s** ,organization**s**, scheme**s** system of rule**s**

After preprocessing

Client
Raam
System
Security

Meaning of all nouns are searched in the WordNet and semantically same nouns are eliminated.

UML Diagram Generation (Cont…)



Requirements Classification

Noun: (ATM, customer, transaction_Information, Bank_computer, account, transaction, bank, time, ATM_network, hours, day, security)

R1: ATM customer **transaction_information** (1,1,1,0,0,0,0,0,0,0,0,0)

R2: **bank_computer transaction_information** ATM account (1,0,1,1,1,0,0,0,0,0,0,0)

R3: **bank_computer transaction_information** ATM transaction (1,0,1,1,0,1,0,0,0,0,0,0)

R4: bank ATMs time (1,0,0,0,0,0,1,1,0,0,0,0)

R5: customer **ATM_network** ATM. (1,1,0,0,0,0,0,0,1,0,0,0)

R6: **customer** ATM (1,1,0,0,0,0,0,0,0,0,0,0)

R7: **ATM_network** hours day (0,0,0,0,0,0,0,0,1,1,1,0)

R8: **ATM_network** security (0,0,0,0,0,0,0,0,1,0,0,1)

Hamming code Distance Measure

R1 = (1,1,1,0,0,0,0,0,0,0,0,0) and
R2 = (1,0,1,1,1,0,0,0,0,0,0,0)
Dis: |R1i − R2i | = |1 − 1| + |1 − 0| + |1 − 1| + |0 − 1| + |0 − 1| + |0 − 0| + |0 − 0| + |0 − 0| + |0 − 0| + |0 − 0| + |0 − 0| + |0 − 0| = 3

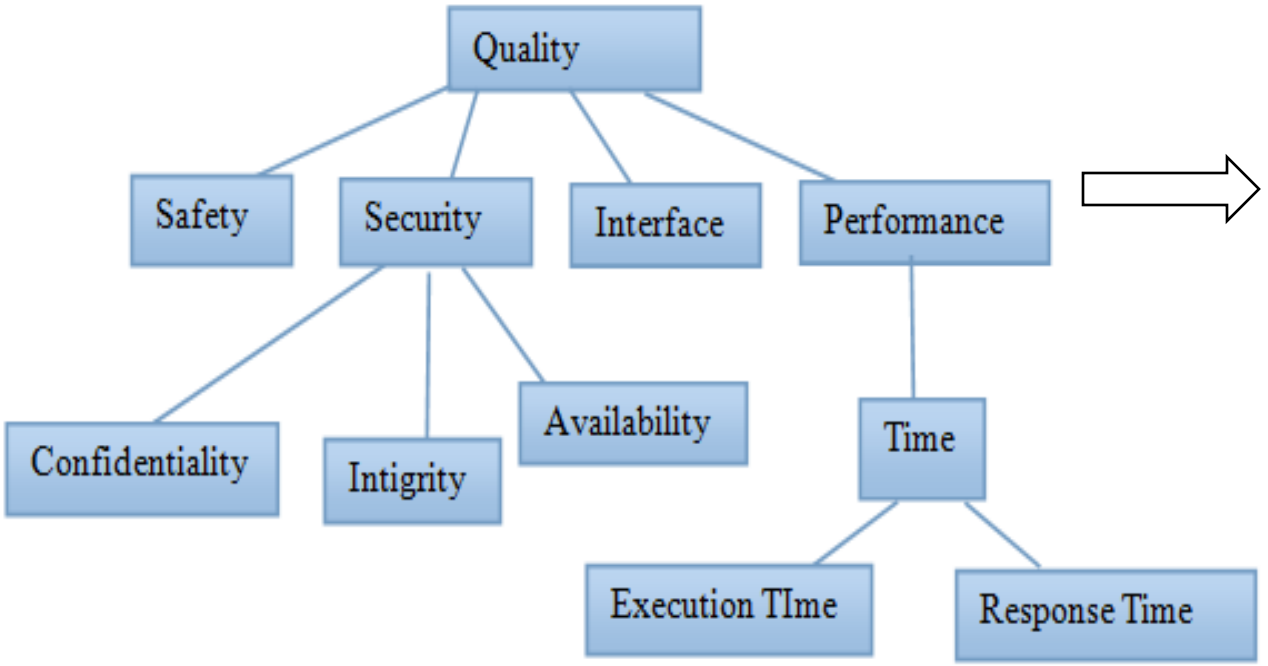
	R1	R2	R3	R4	R5	R6	R7	R8
R1	-	3	3	4	2	1	6	5
R2	3	-	2	5	5	4	7	6
R3	3	2	-	5	5	4	7	6
R4	4	5	5	-	4	3	6	5
R5	2	5	5	4	-	1	4	3
R6	1	4	4	3	1	-	5	4
R7	6	7	7	6	4	5	-	3
R8	5	6	6	5	3	4	3	-

Cluster Formation

R1, R6
R5, R6 -> MERGE : R5 to R1,R6 -> R1, R5, R6
R2, R3
R1, R2 -> MERGE : R1 to R2,R3 -> R1, R2, R3
.....

CLUSTER1:
R1: The ATM interacts with the customer to gather **transaction_information**.
R2: The **bank_computer** gets the **transaction_information** from the ATM to verify an account.
R3: The **bank_computer** gets the **transaction_information** from the ATM to process a transaction.
CLUSTER2:
R4: Each bank may be processing transactions from several ATMs at the same time.
R6: **It** must be very easy for **customer** to use the ATM.
CLUSTER3:
R5: The customer interacts with the **ATM_network** via the ATM.
R7: The **ATM_network** has to be available 24 hours a day.
R8: The **ATM_network** should provide maximal security.

NFRs Elicitation using Quality Ontology



R3: Several ATMs

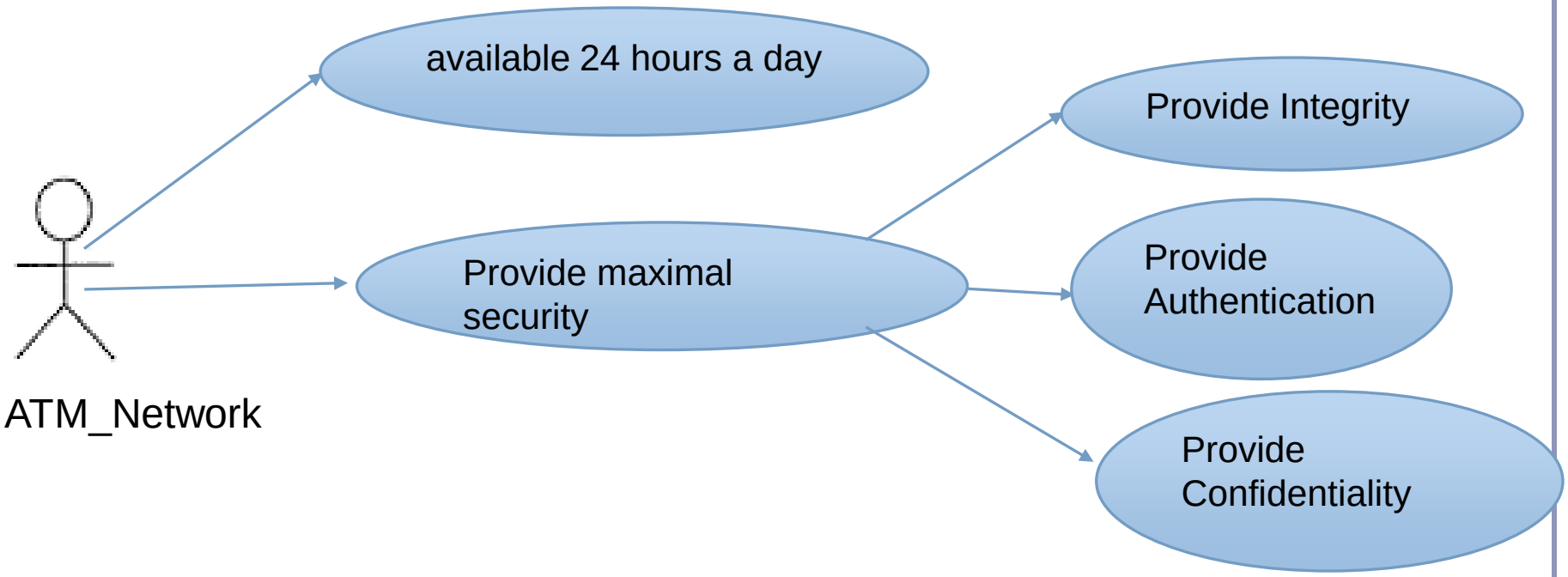
R4: Same time

R6: Very easy for user

R7: Available 24 hours

R8: Maximal security

Extended UML Diagram including NFRs



Conclusions

- We present a hybrid approach that provides a specification of NFRs
- It addresses the problems viz.
    - Normalizing Requirements, Ambiguity Resolving, Requirements Clustering, NFRs Specification
        - Using efficient NLP, a set of rules, ontology and machine learning.
- After analyzing feasibility of the approach on a case study,
- We conclude that the deployment of the approach in the RE practice would have a positive impact

Thank You…