Automated Extraction of Security Policies from Natural-Language Software Documents

Xusheng Xiao¹  Amit Paradkar²
Suresh Thummalapenta³   Tao Xie¹

¹North Carolina State University, Raleigh, NC, USA
²IBM T. J. Watson Research Center, Hawthorne, NY, USA
³IBM Research, Bangalore, India

¹xxiao2@ncsu.edu, ²paradkar@us.ibm.com,
³surthumm@in.ibm.com, ¹xie@csc.ncsu.edu
Textual Artifacts in Software Engineering

- Textual artifacts are prevalent in software engineering.

Software Design Documents

Requirements Specifications

Bug Reports

API Documents
Natural Language Processing (NLP) techniques help computers understand NL artifacts.

In general, NLP is still difficult.

NLP on domain specific sentences with specific styles is feasible.

NL security requirements follow a specific style in describing access control policy (ACP) rules for role-based ACPs.

Example:

```
[subject] [can/cannot/is allowed to] [action] [resource]
for role-based ACPs
```
Access Control Policies (ACP)

- Access control is often governed by security policies called Access Control Policies (ACP)
  - Includes rules to control which principals have access to which resources

ex.

“The Health Care Personnel (HCP) does not have the ability to edit the patient's account.”

- A policy rule includes four elements
  - Subject – HCP
  - Action – edit
  - Resource - patient's account
  - Effect - deny
Improper access control causes problems (e.g., information exposures)
• Incorrect specification
• Incorrect enforcement
Problems of ACP Practice

- In practice, ACPs
  - Buried in requirement documents
  - Written in NL and not checkable

- Functional requirements, such as use cases
  - Consist of a sequence of action steps (ASs)
    - ex. The patient views access log.
  - Imply actors (principals) accessing different resources for achieving some functionalities
  - Written in NL and not checkable
Our Approach: Text2Policy

- NL documents could be large in size
  - Manual extraction is labor-intensive and tedious

- Our approach, Text2Policy
  - Adapt NLP techniques
    - Syntactic and semantic analysis
  - Extract model instances
    - Access Control Policy (ACP) model
    - Action Step (AS) model
  - Produce formal specifications
    - e.g., eXtensible Access Control Markup Language (XACML)
Outline

- Introduction
  - Motivation and problems
- Approach
- Evaluations
- Summary
A HCP should not change patient’s account.

An [subject: HCP] should not [action: change] [resource: patient’s account].

Linguistic Analysis

Model-Instance Construction

Transformation
Linguistic Analysis

- Incorporate syntactic and semantic analysis
  - **syntactic** structure -> noun group, verb group, etc.
  - **semantic** meaning -> subject, action, resource, negative meaning, etc.

- Provide New techniques for model extraction
  - Identify ACP and AS sentences
  - Infer semantic meaning
Common Techniques

- Shallow parsing
- Domain dictionary
- Anaphora resolution

An HCP can view patient’s account. He is disallowed to change the patient’s account.
Technical Challenges (TC) in ACP Extraction

- **TC1: Semantic Structure Variance**
  - different ways to specify the same rule

- **TC2: Negative Meaning Implicitness**
  - verb could have negative meaning
Address TC1 Semantic Structure Variance

Compose pattern based on grammatical function

ex.
An HCP is disallowed to change the patient’s account.

passive voice  \textit{followed by}  to-infinitive phrase
Negative-Expression Identification

- Address TC2 Negative Meaning Implicitness

- Negative expression
  - “not” in subject:
    - ex. No HCP can edit patient’s account.
  - “not” in verb group:
    - ex. HCP can not edit patient’s account.
    - HCP can never edit patient’s account.

- Negative meaning words in main verb group
  - ex. An HCP is disallowed to change the patient’s account.
AS: Syntactic-Pattern Matching

- Syntactic elements
  - Subject, Main verb, Object

- Subject and Object Checking
  - subject is a not a user or object is not a resource

ex. **The prescription list** should include medication, the name of the doctor. . .

- Filtering negative-meaning sentences
  - Negative sentences tend not to describe ASs
A HCP should not change patient’s account.

An [subject: HCP] should not [action: change] [resource: patient’s account].
An HCP is disallowed to change the patient’s account.

- Identify subject, action, and resource:
  - Subject: HCP
  - Action: change
  - Resource: patient’s account

- Infer effect:
  - Negative Expression: none
  - Negative Verb: disallow
  - Inferred Effect: deny
Use case patterns
- industry use cases [DSN’09]
- public use cases

Model-Instance Construction

ex. The patient views access log.
Technical Challenges in Action-Step Extraction

- **AS 1:** He edits the account.
- **AS 2:** The system updates the account.
- **AS 3:** The system displays the updated account.

**HCP:**
- **HCP views the updated account.**

- **TC4:** Transitive Subject
- **TC5:** Perspective Variance
Address TC4 Transitive Subject

Apply data flow to track non-system subject:

AS 1: The HCP edits the account.
AS 2: The system updates the account.

Tracking

Only system as subject

replaced with HCP as subject
Address TC5 Perspective Variance
Apply data flow to track non-system subject:

AS 1: The HCP edits the account.
AS 2: The system shows the updated account.

Tracking

Only system as subject and action is output
Converting to “HCP views the updated account”
RQ1: How effectively does Text2Policy identify ACP sentences in NL documents?

RQ2: How effectively does Text2Policy extract ACP rules from ACP sentences?

RQ3: How effectively does Text2Policy extract action steps from action-step sentences?
Evaluation – Subject

- **iTrust open source project**
  - 448 use-case sentences (37 use cases)
  - preprocessed use cases

- **Collected ACP sentences**
  - 100 ACP sentences
  - From 17 sources (published papers and websites)

- **A module of an IBMAApp (financial domain)**
  - 25 use cases
Apply Text2Policy to identify ACP sentences in iTrust use cases and IBMApp use cases. Text2Policy effectively identifies ACP sentences with precision and recall more than 88%. Precision on IBMApp use cases is better, as proprietary use cases are often of higher quality compared to open-source use cases.
Apply Text2Policy to extract ACP rules from ACP sentences

Text2Policy effectively extracts ACP model instances with accuracy above 86%
Apply Text2Policy to extract action steps from iTrust and IBMApp use cases

<table>
<thead>
<tr>
<th>Subjects</th>
<th># AS Sent.</th>
<th># Extracted</th>
<th>Accu.</th>
</tr>
</thead>
<tbody>
<tr>
<td>iTrust</td>
<td>312</td>
<td>258</td>
<td>82.7%</td>
</tr>
<tr>
<td>IBMApp</td>
<td>455</td>
<td>370</td>
<td>81.3%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>767</strong></td>
<td><strong>628</strong></td>
<td><strong>81.9%</strong></td>
</tr>
</tbody>
</table>

Text2Policy effectively extracts AS model instances with accuracy above 81%

Limitations:
- Subordinate conjunction *or else* and long phrases
Detected Inconsistencies

- No violation between ASs against the extracted ACPs
- Inconsistent names used for referring to the same entity (e.g., user) across different use cases

ex.

*editor* used in UC 4 of iTrust use cases actually refers to *HCP, admin, and all users* in UCs 1, 2, and 4
Natural Language Processing (NLP) for domain-specific purposes is feasible
  - Challenging for general documents
  - Feasible for domain-specific sentences with specific styles

New techniques are required
  - Addressing unique challenges in software engineering

http://research.csc.ncsu.edu/ase/projects/text2policy/
Acknowledgement

- This work is supported in part by NSF grants CCF-0845272, CCF-0915400, CNS-0958235, ARO grant W911NF-08-1-0443, an NSA Science of Security Lablet grant, and a NIST grant.
Related Work

- Manual extraction of ACPs from requirements documents
  - ReCAPS [Inf. Softw. Technol ‘2009]
- Template matching [Artif. Intell’ 2005]
- NLP to analyze API documents
  - Machine learning [ASE’ 2009]
  - Semantic patterns of POS tags [ICSE’ 2012]
- SPARCLE workbench [CHI’ 2006]
- Use case analysis [ICST’ 2010]
Discussion and Future Work

- Construction of complete ACPs
- ACP modeling in the absence of security requirements
- Cooperation between tool and human
- Extended policy extraction
  - ACP-rule ordering
  - Context-aware analysis in action-step extraction
  - Other policy models
    - conditional ACP rules and HIPAA
Transformation – ACP Model

- ACP model
  - XACML policy rules

```
<Rule Effect="deny" RuleId="rule-1">
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="string-equal">
          <AttrValue>HCP</AttrValue>
          <SubjectAttrDesignator AttrId="subject:role"/>
        </SubjectMatch>
      </Subject>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="string-equal">
          <AttrValue>UPDATE</AttrValue>
          <ResourceAttrDesignator AttrId="resource-id"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch MatchId="string-equal">
          <AttrValue>patient.account</AttrValue>
          <ActionAttrDesignator AttrId="action-id"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
</Rule>
```
Transformation – AS Model

- AS model
  - XACML requests

```xml
<Request>
  <Subject>
    <SubjectMatch MatchId="string-equal">
      <AttrValue>HCP</AttrValue>
      <SubjectAttrDesignator AttrId="subject:role"/>
    </SubjectMatch>
  </Subject>
  <Resource>
    <ResourceMatch MatchId="string-equal">
      <AttrValue>account</AttrValue>
      <ResourceAttrDesignator AttrId="resource-id"/>
    </ResourceMatch>
  </Resource>
  <Action>
    <ActionMatch MatchId="string-equal">
      <AttrValue>CREATE</AttrValue>
      <ActionAttrDesignator AttrId="action-id"/>
    </ActionMatch>
  </Action>
</Request>
```
**Example ACPs and Failed Extraction**

- **ACPs**
  - ex. The administrator is not allowed through the system interface to delete an existing entry.

- **Failed extraction**
  - ex. Any subject with an e-mail name in the med.example.com domain can perform any action on any resource.
Example ACPs, FPs and FNs

- **ACPs**
  
  ex. HCPs can modify or delete the fields of the office visit information.

- **FPs**
  
  ex. The instructions can contain numbers, characters. . .

- **FNs**
  
  ex. The LHCP can select a patient to obtain additional information about a patient.
Example Failed Extractions

Failed extractions

ex. The HCP must provide instructions, or else they cannot add the prescription.

ex. The public health agent can send a fake email message to the adverse event reporter to gain more information about the report.