User Guidance for Creating Precise and Accessible Property Specifications

Rachel L. Cobleigh, George S. Avrunin, and Lori A. Clarke

Laboratory for Advanced Software Engineering Research
University of Massachusetts Amherst
http://laser.cs.umass.edu/
Property Specification Problem

- A property focuses on describing one particular aspect of system behavior
  - Even with such focus, it can still be difficult to write a property correctly
- A property should be precise and accessible
  - precise enough to support unambiguous communication and automated analyses
  - accessible enough to be readily understood
After receiving a physician order for a lab test and before obtaining a blood specimen, the nurse must verify that the specimen vial label is correct before labeling the vial.
Our Approach

- Provides property templates that explicitly show subtle variations as options
  - Extends property patterns
    [Dwyer, Avrunin, & Corbett 1998; 1999]

- Provides multiple views of the property
  - Views chosen to support precision, accessibility, and user guidance
  - User can work with one or more of the views
    - Changes made in a view are reflected in the others

- Implemented prototype tool, Propel
Outline

- Background
- Question Tree View
- Evaluation
Transfusion Property

After receiving a physician order for a lab test and before obtaining a blood specimen, the nurse must verify that the specimen vial label is correct before labeling the vial.
After receiving a physician order for a lab test and before obtaining a blood specimen, the nurse must verify that the specimen vial label is correct before labeling the vial.

Events:

• receive-order
• obtain-specimen
• verify-label
• label-vial
Transfusion Property

After receiving a physician order for a lab test and before obtaining a blood specimen, the nurse must verify that the specimen vial label is correct before labeling the vial.

Events:
- receive-order
- obtain-specimen
- verify-label
- label-vial

behavior describes the restrictions on occurrences of events
Transfusion Property

After receiving a physician order for a lab test and before obtaining a blood specimen, the nurse must verify that the specimen vial label is correct before labeling the vial.

Events:
- receive-order
- obtain-specimen
- verify-label
- label-vial

**behavior** describes the restrictions on occurrences of events

**scope** describes the parts of the event sequences within which the behavior restrictions apply
Two Property Views

• Precision: Finite-State Automaton (FSA) template view
  • extends FSA notation

• Accessibility: Disciplined Natural Language (DNL) template view
  • based on natural language
Finite-State Automaton (FSA) Template
Finite-State Automaton (FSA) Template

- (verify-label, label-vial)

verify-label

- label-vial
  or
  verify-label
  or
  - (verify-label, label-vial)

label-vial

- label-vial
  or
  verify-label
  or
  - verify-label
  or
  - (verify-label, label-vial)

- (verify-label, label-vial)
  or
  •
**Disciplined Natural Language (DNL) Template**

`label-vial` cannot occur unless `verify-label` has already occurred.

is not required to occur.

Before the first `verify-label` occurs, the events in the alphabet of this property, other than `label-vial`, can occur any number of times.

After `verify-label` occurs and before the first subsequent `label-vial` occurs:

After the first subsequent `label-vial` occurs:
**Disciplined Natural Language (DNL) Template**

**label-vial** cannot occur unless **verify-label** has already occurred.

**verify-label** is required to occur, but **verify-label** is not required to occur, however

It is acceptable if **verify-label** does not occur, however

After **verify-label** occurs and before the first subsequent **label-vial** occurs:

After the first subsequent **label-vial** occurs:
Propel Templates

<table>
<thead>
<tr>
<th>SCOPES</th>
<th>BEHAVIORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Global</td>
<td>Response</td>
</tr>
<tr>
<td>Before <strong>end</strong></td>
<td><strong>A</strong> results in <strong>B</strong></td>
</tr>
<tr>
<td>After <strong>start</strong></td>
<td>Precedence</td>
</tr>
<tr>
<td><strong>Between start and end</strong></td>
<td><strong>A</strong> enables <strong>B</strong></td>
</tr>
<tr>
<td></td>
<td>Absence</td>
</tr>
<tr>
<td></td>
<td><strong>A</strong> never occurs</td>
</tr>
<tr>
<td></td>
<td>Existence</td>
</tr>
<tr>
<td></td>
<td><strong>A</strong> must occur</td>
</tr>
</tbody>
</table>
Question Tree View

• Problem: users need guidance to choose appropriate scope and behavior

• Question Tree View is designed to provide this guidance
  • One tree for scope and one for behavior

• Question Trees are also useful for resolving detailed options
Behavior Question Tree

• How many events of primary interest are there?
  • One event
  • Two events
Behavior Question Tree

• How many events of primary interest are there?
  • One event
  • Two events
Behavior Question Tree

• How many events of primary interest are there?
  • One event
  • Two events
    • How do \texttt{verify-label} and \texttt{label-vial} interact?
      • \texttt{verify-label} causes \texttt{label-vial} to occur
      • \texttt{label-vial} cannot occur until after \texttt{verify-label} has occurred
Behavior Question Tree

• How many events of primary interest are there?
  • One event
  • Two events
    • How do **verify-label** and **label-vial** interact?
      • verify-label causes **label-vial** to occur
      • **label-vial** cannot occur until after **verify-label** has occurred
Behavior Question Tree

- How many events of primary interest are there?
  - One event
  - Two events
    - How do `verify-label` and `label-vial` interact?
      - `verify-label` causes `label-vial` to occur
        - `label-vial` cannot occur until after `verify-label` has occurred
          - Is `verify-label` required to occur at least once, whether or not `label-vial` eventually occurs?
[insert Propel tool demo here]
Example Completed Behavior

event alphabet: \{verify-label, label-vial, leave-room\}

<table>
<thead>
<tr>
<th>How many events of primary interest are there?</th>
<th>One event</th>
</tr>
</thead>
<tbody>
<tr>
<td>How do verify-label and label-vial interact?</td>
<td></td>
</tr>
<tr>
<td>verify-label causes label-vial to occur</td>
<td></td>
</tr>
<tr>
<td>label-vial cannot occur until after verify-label has occurred</td>
<td></td>
</tr>
<tr>
<td>Is verify-label required to occur at least once?</td>
<td></td>
</tr>
<tr>
<td>Yes, verify-label is required to occur at least once</td>
<td></td>
</tr>
<tr>
<td>No, verify-label is not required to occur at least once</td>
<td></td>
</tr>
<tr>
<td>After verify-label occurs, can verify-label occur again before the first subsequent label-vial occurs?</td>
<td></td>
</tr>
<tr>
<td>Yes, verify-label can occur multiple times before the first subsequent label-vial occurs</td>
<td></td>
</tr>
<tr>
<td>No, verify-label cannot occur again before the first subsequent label-vial occurs</td>
<td></td>
</tr>
<tr>
<td>After verify-label occurs, can events in the alphabet of this property, other than verify-label or label-vial, occur before the first subsequent label-vial occurs?</td>
<td></td>
</tr>
<tr>
<td>Yes, other events in the alphabet of this property can occur before the first subsequent label-vial occurs</td>
<td></td>
</tr>
<tr>
<td>No, other events in the alphabet of this property cannot occur before the first subsequent label-vial occurs</td>
<td></td>
</tr>
<tr>
<td>After the first subsequent label-vial occurs:</td>
<td></td>
</tr>
<tr>
<td>label-vial cannot occur again until after another verify-label occurs, verify-label can occur and if it does, then the situation is the same as when the first verify-label occurred, meaning that all restrictions described on the events will again apply</td>
<td></td>
</tr>
<tr>
<td>both verify-label and label-vial can occur any number of times and do not impose any restrictions on the occurrences of any future events</td>
<td></td>
</tr>
<tr>
<td>verify-label can occur any number of times, but label-vial cannot occur again. Further occurrences of verify-label do not impose additional restrictions on the occurrences of any future events</td>
<td></td>
</tr>
<tr>
<td>label-vial can occur any number of times, but verify-label cannot occur again. Further occurrences of label-vial do not impose additional restrictions on the occurrences of any future events</td>
<td></td>
</tr>
<tr>
<td>neither verify-label nor label-vial can occur again</td>
<td></td>
</tr>
</tbody>
</table>

It is acceptable if verify-label does not occur, however, and if it does not occur then label-vial can never occur. Even if verify-label does occur, label-vial is not required to occur.

Before the first verify-label occurs, the events in the alphabet of this property, other than label-vial, can occur any number of times.

After verify-label occurs and before the first subsequent label-vial occurs:
- no events in the alphabet of this property, other than verify-label, can occur.
- verify-label can occur any number of times

After the first subsequent label-vial occurs:
- the events in the alphabet of this property, other than verify-label or label-vial, could occur any number of times;
- label-vial cannot occur again until after another verify-label occurs;
- verify-label can occur and if it does, then the situation should be regarded as exactly the same as when the first verify-label occurred, meaning that all restrictions described on the events would again apply.
Evaluations

• Used Propel in four real-world case studies

• Completed a small study to see how well people understand the Disciplined Natural Language view
Case Studies

- Four medical safety case studies
  - Blood Transfusion (UMass School of Nursing)
  - Chemotherapy (Baystate Medical Center)
  - Emergency Department (Baystate Medical Center)
  - Blood Bank (Defense Blood Standard System)

- ~80 properties total
Case Studies: Methodology

- Elicited properties from domain experts via interviews or existing documentation

- Elucidated property details:
  - For most properties, used Propel alongside domain experts
  - For a few properties, domain experts used Propel directly

- Domain experts reviewed Propel property specifications and worked with us to improve them
Case Studies: Observations

- Current implementation can express ~80% of the properties

- Cannot yet express:
  - certain property compositions
    - e.g., chaining (6), blocking (3), nested scopes (3)
  - event disjunction/conjunction (3)
  - real-time properties (2)
Case Studies: Observations

- Different distribution of behavior frequencies than in property patterns survey
  [Dwyer et al. 1999]

<table>
<thead>
<tr>
<th></th>
<th>Pattern Survey</th>
<th>Case Studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>44%</td>
<td>21%</td>
</tr>
<tr>
<td>Precedence</td>
<td>5%</td>
<td>63%</td>
</tr>
<tr>
<td>Absence</td>
<td>15%</td>
<td>1%</td>
</tr>
<tr>
<td>Existence</td>
<td>5%</td>
<td>1%</td>
</tr>
</tbody>
</table>

- Roughly the same high percentage of properties are covered
Case Studies: Observations

- Different domain experts were comfortable with different property views
- Asking domain experts to carefully specify subtle details
  - made them aware of common interpretation errors
  - heightened their awareness of safety hazards in practice
  - changed the language they used
  - prompted the creation of new properties
Disciplined Natural Language (DNL) Study

- Completed a small study to see if people interpret the DNL as we intended

- Selected a diverse sample of properties

- Asked participants to translate DNL into FSAs
  - 14 participants: Computer Science graduate students and technical staff
  - Gave each person 1 simple “training” property and 3 more complex properties

- For each translated FSA, estimated how “closely” that FSA and the Propel FSA matched
DNL Study: Observations

• Comparing translated FSAs to Propel FSAs:

<table>
<thead>
<tr>
<th></th>
<th>all FSAs (42)</th>
<th>no Between-scope FSAs (28)</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact match</td>
<td>40%</td>
<td>57%</td>
</tr>
<tr>
<td>“close” match (incl. exact matches)</td>
<td>64%</td>
<td>82%</td>
</tr>
</tbody>
</table>

• It is difficult to clearly express Between scope’s subtle details precisely in natural language

• Participants interpreted most of the DNL the way we intended
Related Work

- Requirements Formalisms
  e.g. Graphical or tabular approaches

- Processing Natural Language (NL) for Requirements Engineering
  e.g. Fuchs, Schwertel, & Schwitter, 1998;
  Gervasi & Zowghi, 2005;
  Breaux, Vail, & Anton, 2006;
  Gervasi & Ambriola, 2006

- Using brief NL notes alongside formal models
  e.g. Dwyer, Avrunin, & Corbett, 1999;
  Drusinsky, 2004;
  Mondragon & Gates, 2004

- Developing NL and formal model in parallel
  e.g. Konrad & Cheng, 2005
Future Work

- Address gaps in Propel expressibility
  - Support both state- and event-based properties
  - Support property compositions

- Provide guidance for how to decompose a property into a behavior and a scope

- Perform more in-depth evaluations of Propel
Summary

• Case studies are ongoing
  • Now ~100 properties

• Initial findings are very promising
  • Good coverage of encountered properties
  • Propel property specifications provide precision and appear to be reasonably accessible
    • Domain experts’ responses are very positive
Thanks!