Specifying process and measuring progress in terms of information state

A.J.J. Dick, M.E.C. Hull and K. Jackson
The Journal of Systems and Software, 2005

2007. 4. 4
Yeongseok Seo
Contents

- Introduction
- Overall approach
- Rule-based process specification
- Implementation in DOORS
- Case study
- Conclusion
- Discussion
Introduction (1/2)

- Trends
  - Project management is considered to be an activity which measures progress in terms of cost and time expended.

![Classic management triangle diagram](image)

Quality of the products produced by each activity is also crucial.
Introduction (2/2)

- **Suggestion**
  - Each activity can be specified in terms of the information that it should generate

  **Using rule-based method**

  - Measure progress
  - Assess the impact of proposed changes
  - Define metrics on how a project is performing
Overall approach

[Specifying process]
Generate requirement process
(Statements, Container, Relationships)

[Modeling information]
Express the kind of information which exists and the relationships

[Measuring progress]
Measure progress by counting the number of rule applicable instances

[Information rules]
Take dynamic constraints on the information model
Specifying process

- Consider three kinds of information
  - Statement (or object), Containers, Relationships
- Consider ‘Generate requirements’ process
  - System requirement document (SRD) is written in response to a supplied user requirement document (URD) for a particular product
  - The end goal is to have responded to every statement in the URD by one or more linked statements in the SRD
Rule-based process specification(2/7)

- Specifying process (cont’d)
  - Consider *Generate requirements* process (cont’d)
    - **Create user requirements**
      - **Create URD**
        - Rule A
        - Rule B
    - **Create system requirements**
      - **Create SRD**
        - Rule C
        - Rule D
      - **Respond to URD**
        - Rule E
    - **Review satisfaction**
      - **Review URD**
        - Rule F

Specify the goal of having complete coverage of the user requirements by system requirements.
Rule-based process specification (3/7)

- Modeling information
  - Express static constraints
  - Express process in terms of the manipulation of data
    - Using UML

Summarize how it is satisfied by the system requirements that respond to it

- User requirement
  + Argument

- System requirement
  + SR_Type
  has transaction

Transaction, Functional and Timing etc
Rule-based process specification (4/7)

- Information rules
  - Express dynamic constraints
  - Have two component
    - Applicability
      - Determined by scope and pre-condition
    - Conformity
      - Determined by post-condition
Rule-based process specification (5/7)

- Information rules (cont’d)
  - Example
    - Every UR in the URD, there should exist one or more SR in the SRD in the ‘responds to’ relationship (Rule E)

RULE E:
Applicability:
ALL UR of type User requirement IN URD
Conformity:
UR. Argument ≠ EMPTY
AND EXISTS SR of type System requirement IN SRD
WHERE SR responds to UR

RULE G:
Applicability:
ALL S of type System Requirement IN SRD
WHERE S. SR_Type=Transaction
Conformity:
EXIST T of type System Requirement IN SRD
WHERE S has transaction T
AND T. SR_Type IN [Functional, Timing]

- Every SR of type Transaction must have at least one has transaction link to another SR of type Functional or Timing
Measuring progress
- Actions will have to be carried out for each UR to create a response in the SRD
  - Outcome should be as specified in Rule E: Every user requirement will have a response
- There are many combinations of actions that would achieve result

1. Select a UR
2. Complete the Argument attribute of the UR
3. Create and write as many SRs as necessary to meet the argument
4. Link these SR to the UR in the ‘responds to’ relationship
Rule-based process specification (7/7)

- Measuring progress
  - Progress of rule-based processes can be measured

\[
\text{progress} = \frac{AC}{A} \quad \text{The number of instances in which the rule applies and shows conformity}
\]

At the start, though the rule applies to all URs, none of the URs conform to the rule: 0%
At the end, the rule shows conformity on every user requirement, and so progress: 100%
Implementation in DOORS(1/5)

- Basic structure of DOORS(v5.1)
  - Project – contain any number of modules
    - Formal module – documents
    - Link module – relationships

- Information model hierarchy in DOORS

  ![Information Model Diagram]

  - Information Model
  - Modules
    - Module 1 ---- Module n
      - Attribute 1 ---- Attribute n
      - Relationships
      - Rules
  - Common Attributes
    - Attribute 1 ---- Attribute n
    - Link Verification Verification
  - Common Rules

Common to all module
- scope: cover the same level of modules
Information model in DOORS

- Requirements modules
  - Common attributes
    - ID
    - Text
    - Date reviewed
    - Status
  - User requirements
    - UR object type
    - Stakeholder
    - Value
  - System requirements
    - SR object type
    - Actor
  - IF Parent Object Type
    - IS Non-Requirement
  - IF Object Type
    - IS Transaction
    - IS Mode
    - IS Timing
  - Traceability->User requirements
  - Traceability->System requirements
  - Transactions->System requirements

Project
Module/relationship
Attribute
Comment
Rule
Implementation in DOORS (3/5)

- Representing rules in DOORS
  - Example rule with executable code fragments

```
IF Object Type
IS Transaction
IS Mode
IS Timing
```

Hierarchy
Rule
Executable code

<table>
<thead>
<tr>
<th></th>
<th>IF Object Type</th>
<th>IS Transaction</th>
<th>IS Mode</th>
<th>IS Timing</th>
<th>return probeAttr(obj, &quot;SR Object Type&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>IF Object Type</td>
<td>IS Transaction</td>
<td>IS Mode</td>
<td>IS Timing</td>
<td>return count_links(obj, &quot;out&quot;, &quot;Traceability&quot;, &quot;System Requirements&quot;, &quot;SR Object Type&quot;, &quot;Mode&quot;) &gt;= 1</td>
</tr>
<tr>
<td>6</td>
<td>IF Object Type</td>
<td>IS Transaction</td>
<td>IS Mode</td>
<td>IS Timing</td>
<td>return count_links(obj, &quot;out&quot;, &quot;Traceability&quot;, &quot;System Requirements&quot;, &quot;SR Object Type&quot;, &quot;Timing&quot;) &gt;= 1</td>
</tr>
<tr>
<td>7</td>
<td>THEN Should have 1 or more Traceability links to a Mode Requirement</td>
<td>IS Transaction</td>
<td>IS Mode</td>
<td>IS Timing</td>
<td>return count_links(obj, &quot;out&quot;, &quot;Traceability&quot;, &quot;System Requirements&quot;, &quot;SR Object Type&quot;, &quot;Mode&quot;) &gt;= 1</td>
</tr>
<tr>
<td>7</td>
<td>THEN Should have 1 or more Traceability links to a Timing Requirement</td>
<td>IS Transaction</td>
<td>IS Mode</td>
<td>IS Timing</td>
<td>return count_links(obj, &quot;out&quot;, &quot;Traceability&quot;, &quot;System Requirements&quot;, &quot;SR Object Type&quot;, &quot;Timing&quot;) &gt;= 1</td>
</tr>
<tr>
<td>7</td>
<td>THEN Should have 1 or more Traceability links to a Capability/Constraint UR</td>
<td>IS Transaction</td>
<td>IS Mode</td>
<td>IS Timing</td>
<td>return count_links(obj, &quot;out&quot;, &quot;Traceability&quot;, &quot;User Requirements&quot;, &quot;UR Object Type&quot;, &quot;(Capability/Constraint)&quot;) &gt;= 1</td>
</tr>
<tr>
<td>7</td>
<td>THEN Should have 1 or more Design links from a Software Design</td>
<td>IS Transaction</td>
<td>IS Mode</td>
<td>IS Timing</td>
<td>return count_links(obj, &quot;in&quot;, &quot;Design Links&quot;, &quot;Design&quot;, &quot;Des Object Type&quot;, &quot;Software&quot;) &gt;= 1</td>
</tr>
</tbody>
</table>
Information model validation

- Tool has been developed which checks conformance of the information against an information model
  - Project level
    - Check missing modules and/or surplus modules
  - Module level
    - Check missing attributes and/or extra attributes
  - Object level
    - Check dynamic conformance of data in a given module against executable rules
Information model validation (cont’d)

- Rule validation results at object level

<table>
<thead>
<tr>
<th>SR Object Type</th>
<th>System requirements for the project</th>
<th>Rule Result</th>
<th>Rule Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>b) night mode. This mode represents the operation of the system for the silent nights.</td>
<td>PASS</td>
<td>[Data-919] IF Object Type/IS Mode/THEN Should have 1 or more Traceability links to a Constraint UR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Data-972] IF Object Type/IS Mode/THEN Should have 1 or more Traceability links from a Hardware Constraint SR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Data-973] IF Object Type/IS Mode/THEN Should have 1 or more Traceability links from a Transaction SR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Data-907] Should not have undocumented links</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAIL</td>
<td>[Data-907] Should not have undocumented links</td>
</tr>
<tr>
<td>Heading</td>
<td>3.3 Operational Requirements</td>
<td>PASS</td>
<td>[Data-784] IF Object Type/IS Transaction/THEN Should have 1 or more Traceability links to a Mode Requirement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Data-917] IF Object Type/IS Transaction/THEN Should have 1 or more Traceability links to a Timing Requirement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Data-974] IF Object Type/IS Transaction/THEN Should have 1 or more Traceability links to a Capability/Constraint UR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Data-979] IF Object Type/IS Transaction/THEN Should have 1 or more Design links from a Software Design</td>
</tr>
<tr>
<td>Transaction</td>
<td>The new system shall be shared among several clients</td>
<td>FAIL</td>
<td>[Data-907] Should not have undocumented links</td>
</tr>
</tbody>
</table>
Case study

- Diverse case studies from partner organizations in an EC funded project
  - Principles in this paper are equally applicable to other problem domains and activities
Conclusion

- **Contribution**
  - Define rule and measure progress
    - Can be automated
  - Provide process guidance
    - Can be guided what actions need to be performed to complete activities
  - Complement other project management tools

- **Future work**
  - Sequencing of activities or tasks
  - Aggregation of progress
Discussion

- Characteristic
  - Progress measuring can consider correlations among tasks using rules

- Critique
  - Total number of instances in which a rule applies can not be known until the process has finished
    - Need estimation