



Dependency Management for System Specification

Dr. Oskar v. Dungern
2014-05-09

Introduction

It is shown how relations between revisions of system components and revisions of requirements are managed.

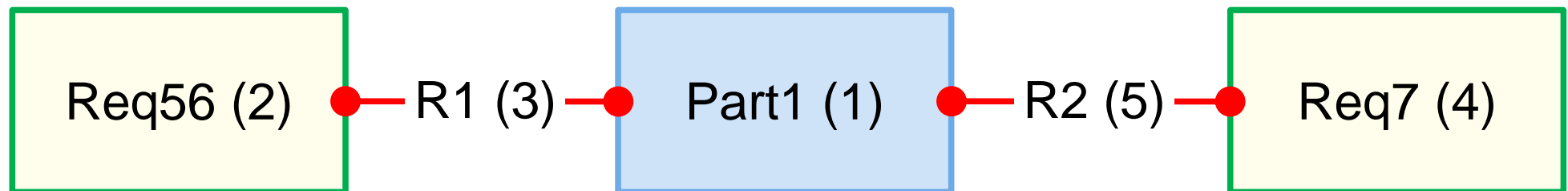
- First, a model is created consisting of
 - objects representing the system components,
 - others representing the requirements and
 - the relations between specified revisions of system components and requirements.
- To alleviate searching and management, an object usually carries a number of attributes (description, category, status,), but this is not considered, here.
- On either side, a relation may be
 - ,floating‘: this end points always to the newest revision of the respective object.
 - ◆ ,fixed‘: this end continues to point to a specified revision of an object, even if the object is updated, i.e. a new revision is created.

1. A system component satisfies two requirements

1.1 Initial Situation

Assume that a system component Part1 must satisfy two relations Req7 and Req56:

- Relation R1 is unspecific to the revisions of both Part1 and Req56 (,floating‘).
- Relation R2 is unspecific to the revisions of both Part1 and Req7 (,floating‘).
- The number in brackets is the transaction count = revision number.

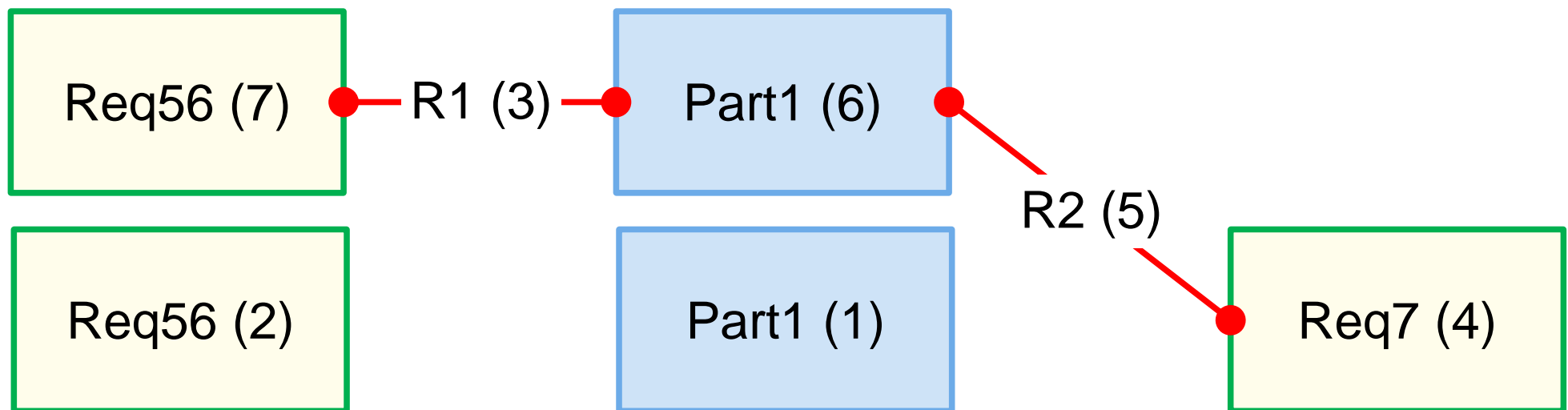


1. A system component satisfies two requirements

1.2 Update the system component and one of the requirements

Suppose, Part1 and Req56 have been updated:

- A new revision of Part1 and a new revision of Req56 are created
- Both relations automatically apply to the newest revisions of the objects
- Without a change to the relations, the requirements that must be supported by Part1 (6) can be listed.

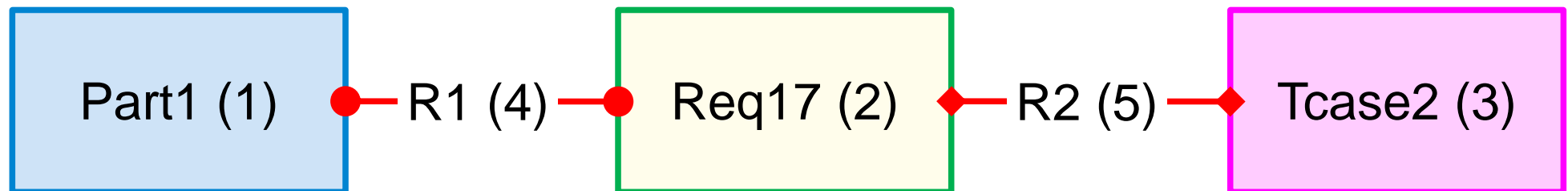


2. A requirement has relations to a component and to a test-case

2.1 Initial Situation

Assume that a requirement has 2 relations:

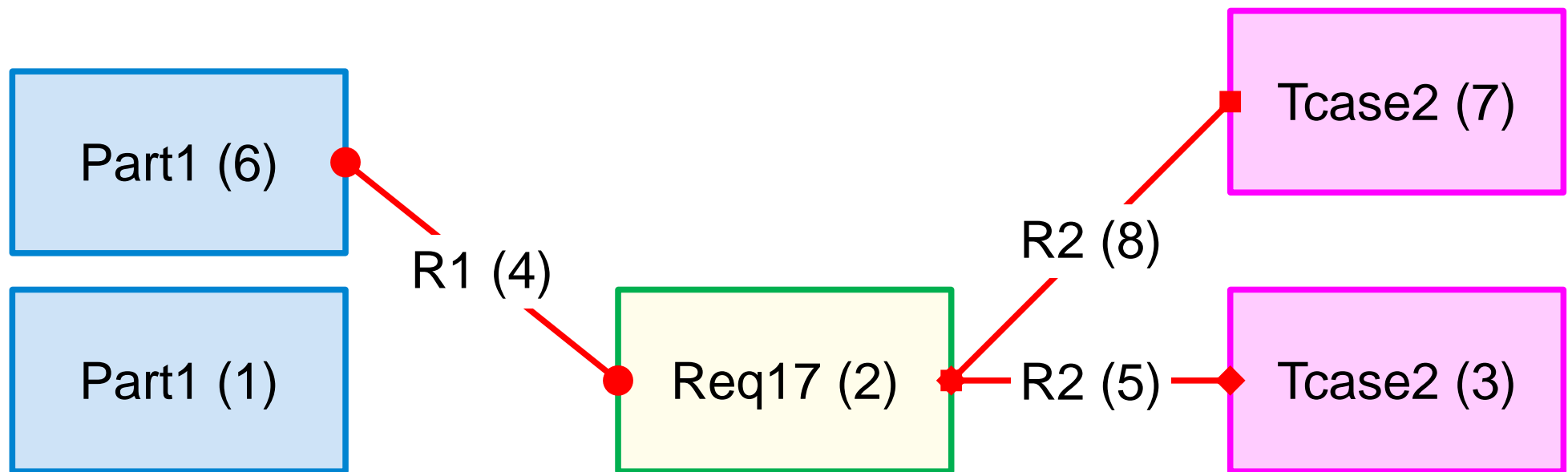
- Part1 ,satisfies‘ Req17: R1 is unspecific to the revisions of both Part1 and Req17 (,floating‘).
- Tcase2 ,verifies‘ Req17: R2 is specific to the revisions of Req17 and Tcase2 (,fixed‘).
- The number in brackets is the transaction count = revision number.



2. A requirement has relations to a component and to a test-case

2.2 Updating Part1 and Tcase2

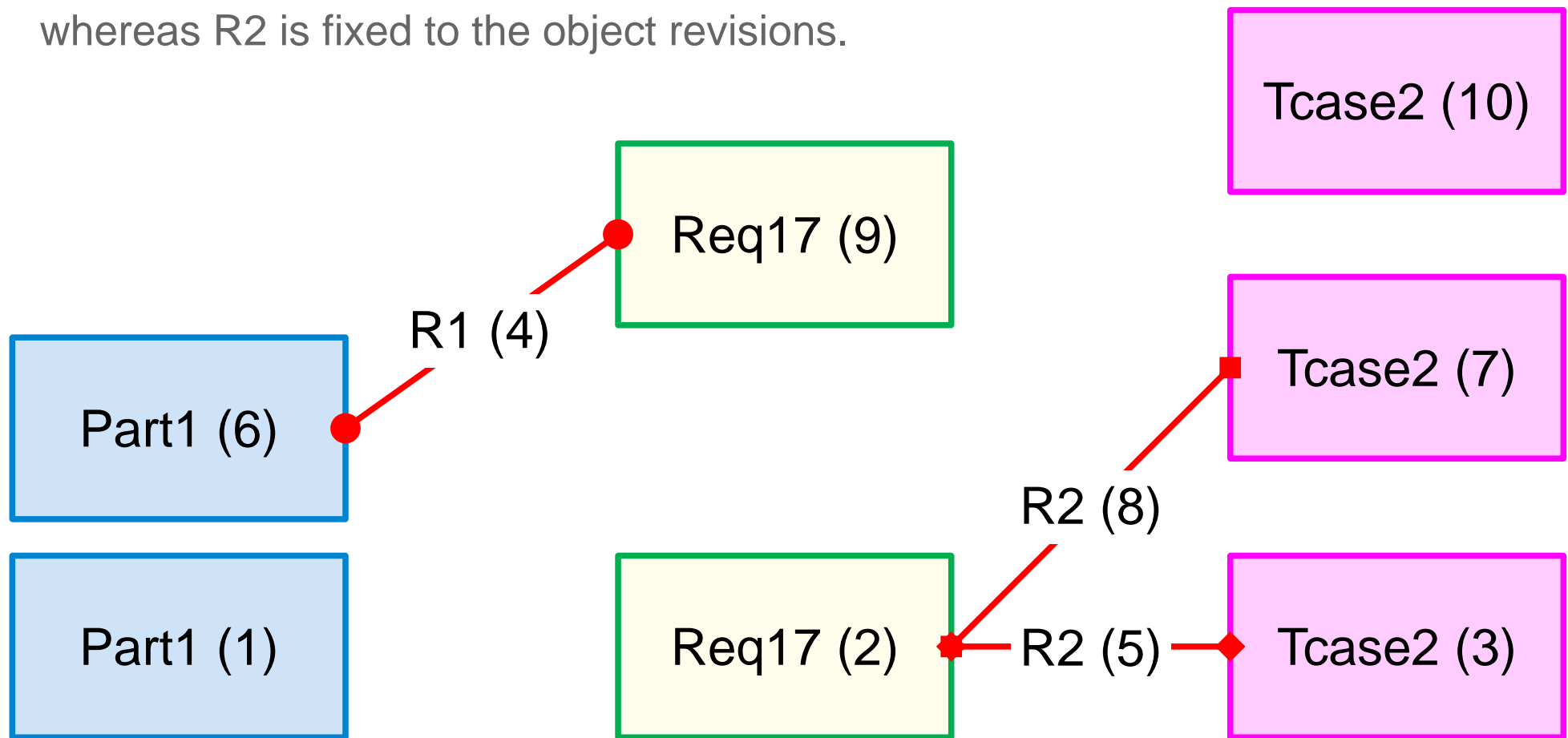
- New revisions of both Part1 and Tcase2 are created.
- Being ,floating‘, relation R1 automatically applies to the new revision of Part1.
- Being ,fixed‘, relation R2 is explicitly updated to connect Req17 to Tcase2 (if appropriate).



2. A requirement has relations to a component and to a test-case

2.3 Updating Req17 and Tcase2

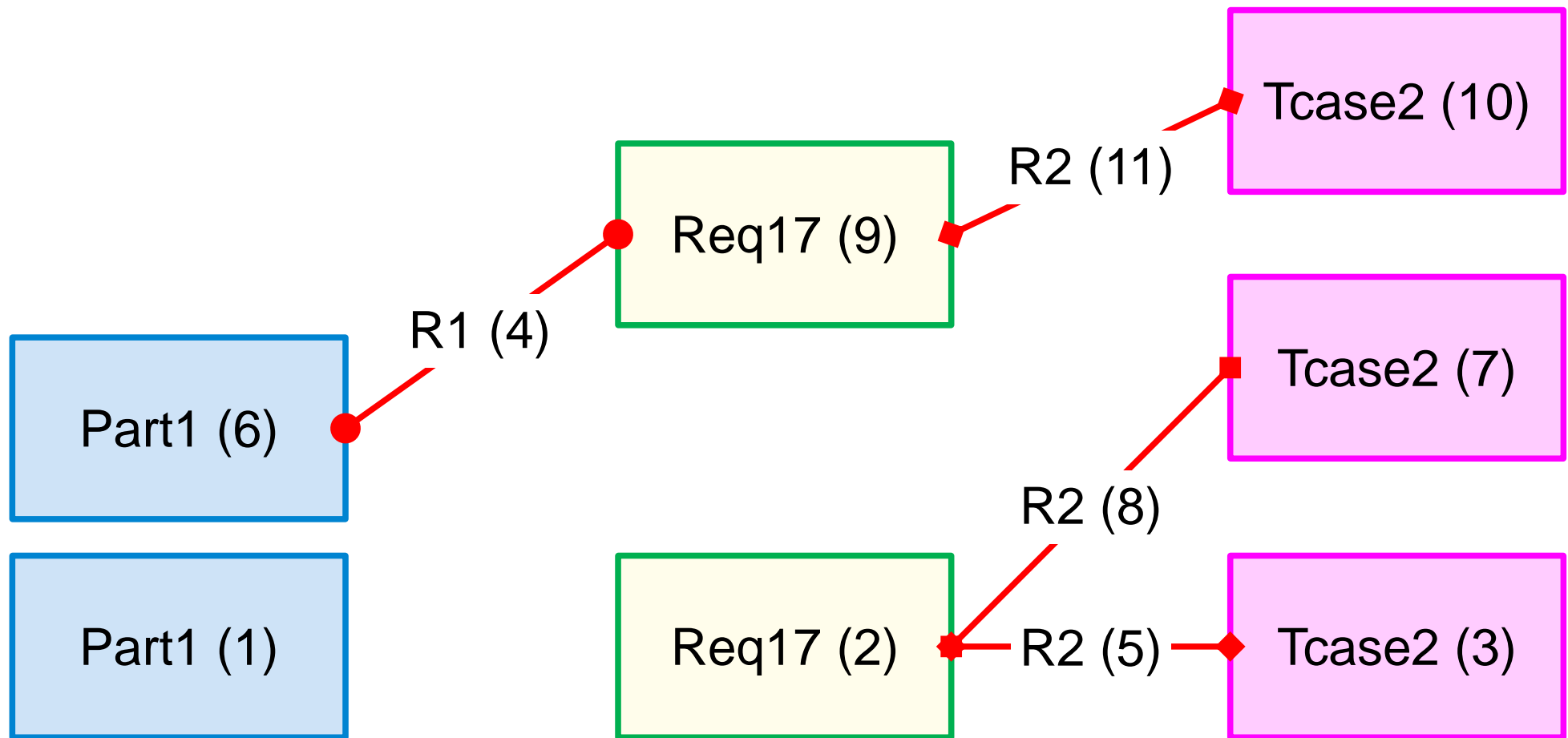
- Also here, being floating, relation R1 automatically applies to the newest revisions of the updated objects,
- whereas R2 is fixed to the object revisions.



2. A requirement has relations to a component and to a test-case

2.4 Updating R2

- Being ,fixed‘, R2 must be explicitly updated to point to the newest revisions.





Is this interesting for you?

 Any questions?

Contact

Dr.-Ing. Oskar v. Dungern
+49 173 670 9958
od@enso-managers.com

Information

www.reqf.de