Characterization of Aggregation and Composition with Object Diagrams

Martin Gogolla

2009-06-19

[2007-11-05]

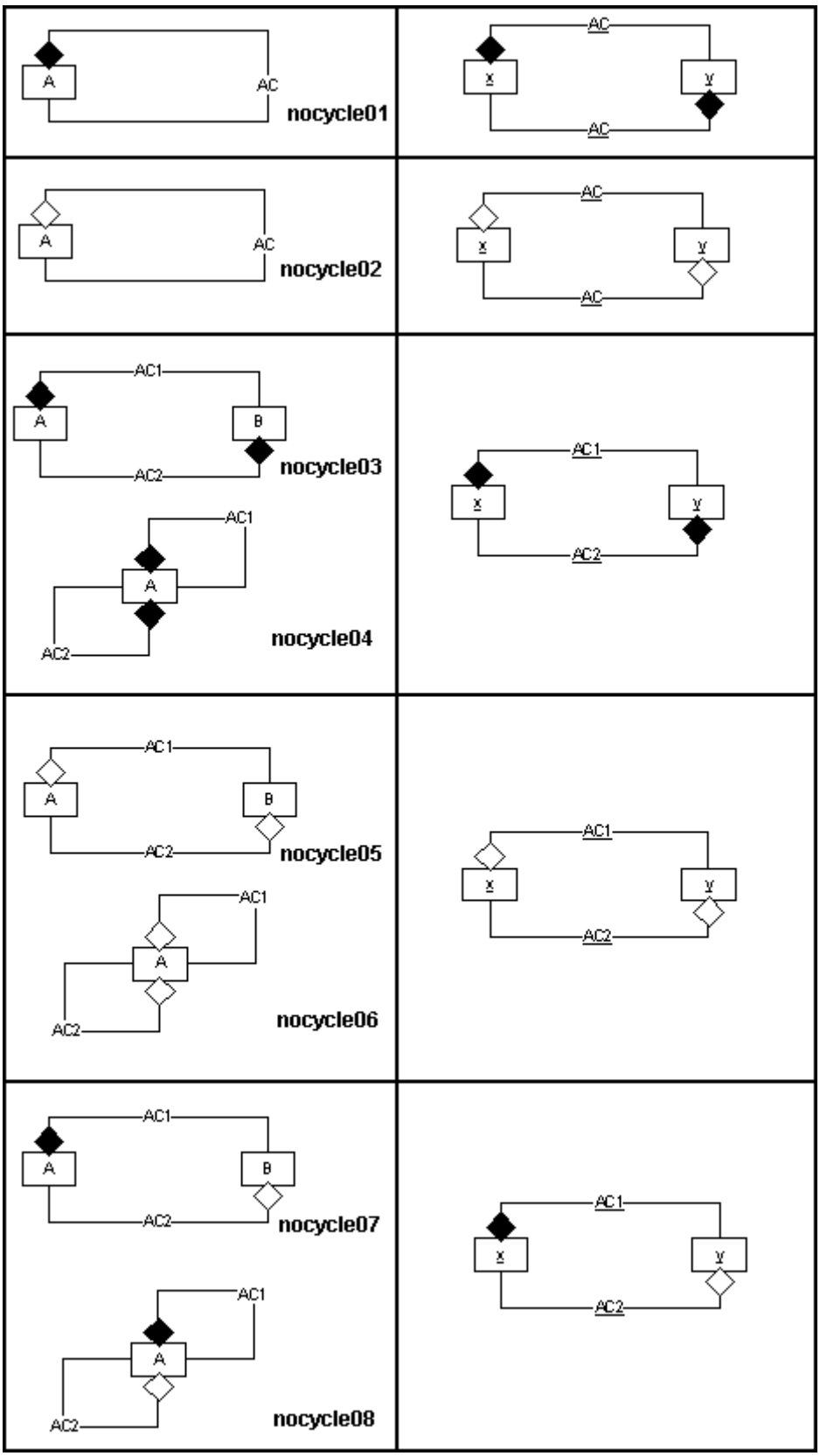
We characterize UML's aggregation and composition with object diagrams. The basic requirements are:

(1) no acycles for aggregation and composition, if one considers all links which exist for a given class diagram (even beyond a single aggregation or composition);

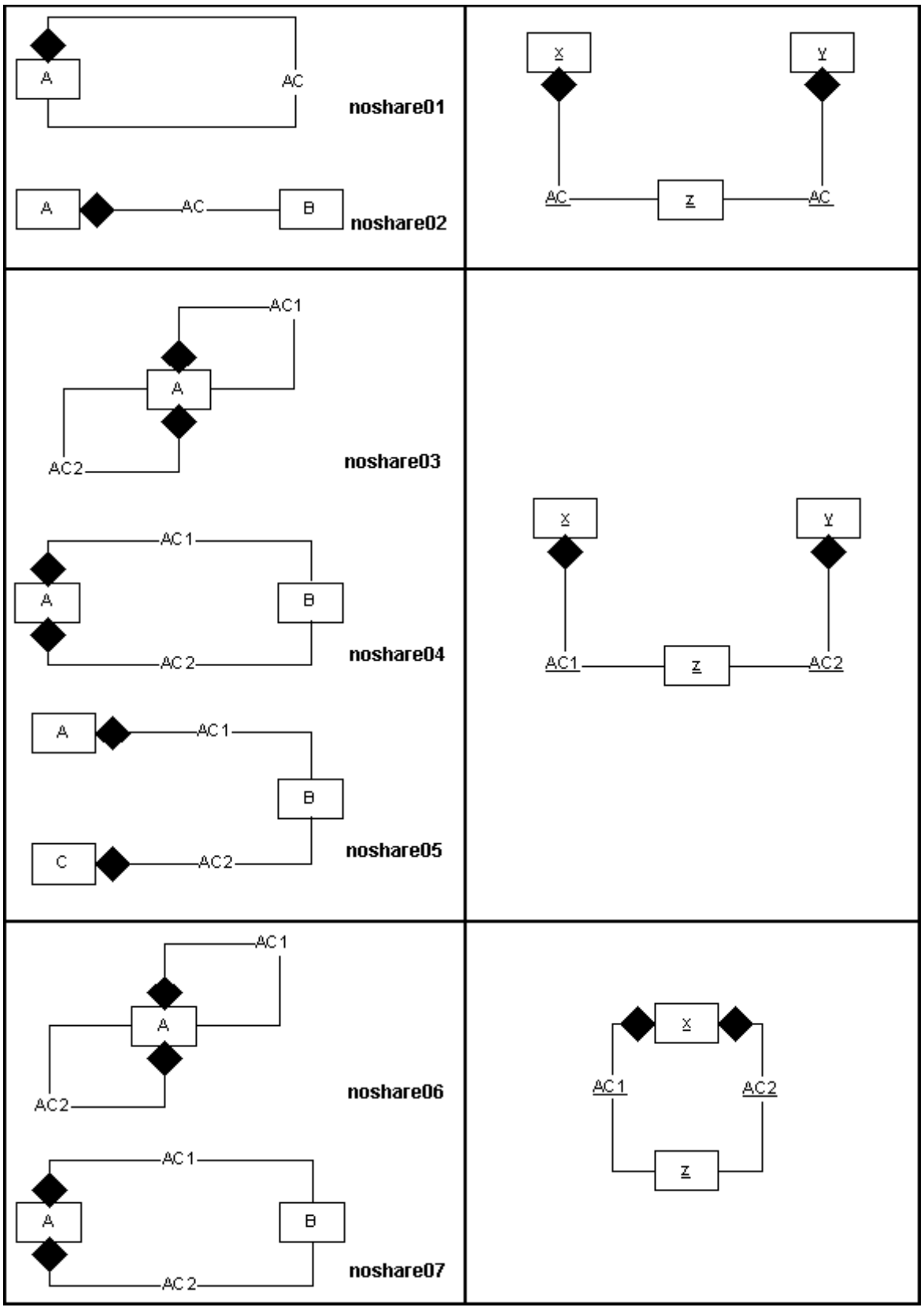
(2) no sharing for composition: an object is allowed to be connected to another object by only one black diamond, i.e., one object can only live via a black diamond in at most one other object (even beyond a single aggregation or composition).

The following two figures explain this with class diagrams on the left and forbidden object diagrams on the right.

Remark on transitivity: All connections (associations or links) in the diagrams are transitive. For the acyclicity, a shown connection may consist of more than one connection consisting of black or white diamond connections. For forbidding of sharing, a shown connection may consist of more that one connection consisting of black diamond connections only.



nocycle01.use, nocycle01.cmd
...
nocycle08.use, nocycle08.cmd



noshare01.use, noshare01.cmd
...
noshare07.use, noshare07.cmd

ERROR MESSAGES - NO CYCLE (insert (y,x))

Error: Insert would result in a cycle in the part-whole hierarchy. Object `y' is a direct or indirect part of `x'.

INSTEAD OF

Error: Detected a cycle in aggregation hierarchy. Object `y' is already a part of `x'.

ERROR MESSAGES - NO SHARING (insert (_,y))

Error: Insert would result in two aggregates for object `y'. Object `y' is already component of another object.

INSTEAD OF

Error: Object `y' is already a part of other composition.

```
--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

composition AC between
  A[0..*] role parent
  A[0..*] role child
end

open nocycle01.use

!create x:A
!insert (x,x) into AC
!create y:A
!insert (x,y) into AC
!insert (y,x) into AC

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

aggregation AC between
  A[0..*] role parent
  A[0..*] role child
end

open nocycle02.use

!create x:A
!insert (x,x) into AC
!create y:A
!insert (x,y) into AC
!insert (y,x) into AC

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

class B
end

composition AC1 between
  A[0..*] role parent1
  B[0..*] role child1
end

composition AC2 between
  B[0..*] role parent2
  A[0..*] role child2
end
```

```
open nocycle03.use

!create x:A
!create y:B
!insert (x,y) into AC1
!insert (y,x) into AC2

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

composition AC1 between
  A[0..*] role parent1
  A[0..*] role child1
end

composition AC2 between
  A[0..*] role parent2
  A[0..*] role child2
end

open nocycle04.use

!create x:A
!create y:A
!insert (x,y) into AC1
!insert (y,x) into AC2

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

class B
end

aggregation AC1 between
  A[0..*] role parent1
  B[0..*] role child1
end

aggregation AC2 between
  B[0..*] role parent2
  A[0..*] role child2
end

open nocycle05.use

!create x:A
!create y:B
!insert (x,y) into AC1
!insert (y,x) into AC2

--------------------------------------------------------------------------
```

```
model TestAggregationComposition

class A
end

aggregation AC1 between
  A[0..*] role parent1
  A[0..*] role child1
end

aggregation AC2 between
  A[0..*] role parent2
  A[0..*] role child2
end

open nocycle06.use

!create x:A
!create y:A
!insert (x,y) into AC1
!insert (y,x) into AC2
```

```
model TestAggregationComposition

class A
end

class B
end

composition AC1 between
  A[0..*] role parent1
  B[0..*] role child1
end

aggregation AC2 between
  B[0..*] role parent2
  A[0..*] role child2
end

open nocycle07.use

!create x:A
!create y:B
!insert (x,y) into AC1
!insert (y,x) into AC2
```

```
model TestAggregationComposition

class A
end

composition AC1 between
  A[0..*] role parent1
  A[0..*] role child1
end

aggregation AC2 between
  A[0..*] role parent2
  A[0..*] role child2
end


open nocycle08.use

!create x:A
!create y:A
!insert (x,y) into AC1
!insert (y,x) into AC2

--------------------------------------------------------------------------
```

```
use> read nocycle01.cmd
nocycle01.cmd> open nocycle01.use
nocycle01.cmd>
nocycle01.cmd> !create x:A
nocycle01.cmd> !insert (x,x) into AC
               Error: Object `x' cannot be a part of itself.
nocycle01.cmd> !create y:A
nocycle01.cmd> !insert (x,y) into AC
nocycle01.cmd> !insert (y,x) into AC
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.

use> read nocycle02.cmd
nocycle02.cmd> open nocycle02.use
nocycle02.cmd>
nocycle02.cmd> !create x:A
nocycle01.cmd> !insert (x,x) into AC
               Error: Object `x' cannot be a part of itself.
nocycle02.cmd> !create y:A
nocycle02.cmd> !insert (x,y) into AC
nocycle02.cmd> !insert (y,x) into AC
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.

use> read nocycle03.cmd
nocycle03.cmd> open nocycle03.use
nocycle03.cmd>
nocycle03.cmd> !create x:A
nocycle03.cmd> !create y:B
nocycle03.cmd> !insert (x,y) into AC1
nocycle03.cmd> !insert (y,x) into AC2
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.

use> read nocycle04.cmd
nocycle04.cmd> open nocycle04.use
nocycle04.cmd>
nocycle04.cmd> !create x:A
nocycle04.cmd> !create y:A
nocycle04.cmd> !insert (x,y) into AC1
nocycle04.cmd> !insert (y,x) into AC2
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.

use> read nocycle05.cmd
nocycle05.cmd> open nocycle05.use
nocycle05.cmd>
nocycle05.cmd> !create x:A
nocycle05.cmd> !create y:B
nocycle05.cmd> !insert (x,y) into AC1
nocycle05.cmd> !insert (y,x) into AC2
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.
```

```
use> read nocycle06.cmd
nocycle06.cmd> open nocycle06.use
nocycle06.cmd>
nocycle06.cmd> !create x:A
nocycle06.cmd> !create y:A
nocycle06.cmd> !insert (x,y) into AC1
nocycle06.cmd> !insert (y,x) into AC2
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.

use> read nocycle07.cmd
nocycle07.cmd> open nocycle07.use
nocycle07.cmd>
nocycle07.cmd> !create x:A
nocycle07.cmd> !create y:B
nocycle07.cmd> !insert (x,y) into AC1
nocycle07.cmd> !insert (y,x) into AC2
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.

use> read nocycle08.cmd
nocycle08.cmd> open nocycle08.use
nocycle08.cmd>
nocycle08.cmd> !create x:A
nocycle08.cmd> !create y:A
nocycle08.cmd> !insert (x,y) into AC1
nocycle08.cmd> !insert (y,x) into AC2
               Error: Detected a cycle in aggregation hierarchy.
               Object `y' is already a part of `x'.
```

```
--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

composition AC between
  A[0..*] role parent
  A[0..*] role child
end

open noshare01.use

!create x:A
!create y:A
!create z:A
!insert (x,z) into AC
!insert (y,z) into AC

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

class B
end

composition AC between
  A[0..*] role parent
  B[0..*] role child
end

open noshare02.use

!create x:A
!create y:A
!create z:B
!insert (x,z) into AC
!insert (y,z) into AC

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

composition AC1 between
  A[0..*] role parent1
  A[0..*] role child1
end

composition AC2 between
  A[0..*] role parent2
  A[0..*] role child2
end
```

```
open noshare03.use

!create x:A
!create y:A
!create z:A
!insert (x,z) into AC1
!insert (y,z) into AC2

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

class B
end

composition AC1 between
  A[0..*] role parent1
  B[0..*] role child1
end

composition AC2 between
  A[0..*] role parent2
  B[0..*] role child2
end

open noshare04.use

!create x:A
!create y:A
!create z:B
!insert (x,z) into AC1
!insert (y,z) into AC2

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

class B
end

class C
end

composition AC1 between
  A[0..*] role parent1
  B[0..*] role child1
end

composition AC2 between
  C[0..*] role parent2
  B[0..*] role child2
end
```

```
open noshare05.use

!create x:A
!create y:C
!create z:B
!insert (x,z) into AC1
!insert (y,z) into AC2

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

composition AC1 between
  A[0..*] role parent1
  A[0..*] role child1
end

composition AC2 between
  A[0..*] role parent2
  A[0..*] role child2
end

open noshare06.use

!create x:A
!create z:A
!insert (x,z) into AC1
!insert (x,z) into AC2

--------------------------------------------------------------------------

model TestAggregationComposition

class A
end

class B
end

composition AC1 between
  A[0..*] role parent1
  B[0..*] role child1
end

composition AC2 between
  A[0..*] role parent2
  B[0..*] role child2
end

open noshare07.use

!create x:A
!create z:B
!insert (x,z) into AC1
!insert (x,z) into AC2

--------------------------------------------------------------------------
```

```
use> read noshare01.cmd
noshare01.cmd> open noshare01.use
noshare01.cmd>
noshare01.cmd> !create x:A
noshare01.cmd> !create y:A
noshare01.cmd> !create z:A
noshare01.cmd> !insert (x,z) into AC
noshare01.cmd> !insert (y,z) into AC
               Error: Object `z' is already a part of other composition.

use> read noshare02.cmd
noshare02.cmd> open noshare02.use
noshare02.cmd>
noshare02.cmd> !create x:A
noshare02.cmd> !create y:A
noshare02.cmd> !create z:B
noshare02.cmd> !insert (x,z) into AC
noshare02.cmd> !insert (y,z) into AC
               Error: Object `z' is already a part of other composition.

use> read noshare03.cmd
noshare03.cmd> open noshare03.use
noshare03.cmd>
noshare03.cmd> !create x:A
noshare03.cmd> !create y:A
noshare03.cmd> !create z:A
noshare03.cmd> !insert (x,z) into AC1
noshare03.cmd> !insert (y,z) into AC2
               Error: Object `z' is already a part of other composition.

use> read noshare04.cmd
noshare04.cmd> open noshare04.use
noshare04.cmd>
noshare04.cmd> !create x:A
noshare04.cmd> !create y:A
noshare04.cmd> !create z:B
noshare04.cmd> !insert (x,z) into AC1
noshare04.cmd> !insert (y,z) into AC2
               Error: Object `z' is already a part of other composition.

use> read noshare05.cmd
noshare05.cmd> open noshare05.use
noshare05.cmd>
noshare05.cmd> !create x:A
noshare05.cmd> !create y:C
noshare05.cmd> !create z:B
noshare05.cmd> !insert (x,z) into AC1
noshare05.cmd> !insert (y,z) into AC2
               Error: Object `z' is already a part of other composition.

use> read noshare06.cmd
noshare06.cmd> open noshare06.use
noshare06.cmd>
noshare06.cmd> !create x:A
noshare06.cmd> !create z:A
noshare06.cmd> !insert (x,z) into AC1
noshare06.cmd> !insert (x,z) into AC2
               Error: Object `z' is already a part of other composition.
```

```
use> read noshare07.cmd
noshare07.cmd> open noshare07.use
noshare07.cmd>
noshare07.cmd> !create x:A
noshare07.cmd> !create z:B
noshare07.cmd> !insert (x,z) into AC1
noshare07.cmd> !insert (x,z) into AC2
               Error: Object `z' is already a part of other composition.
```