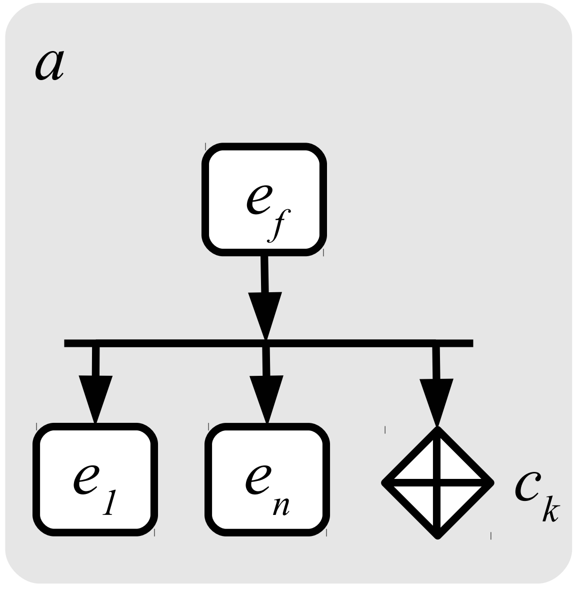
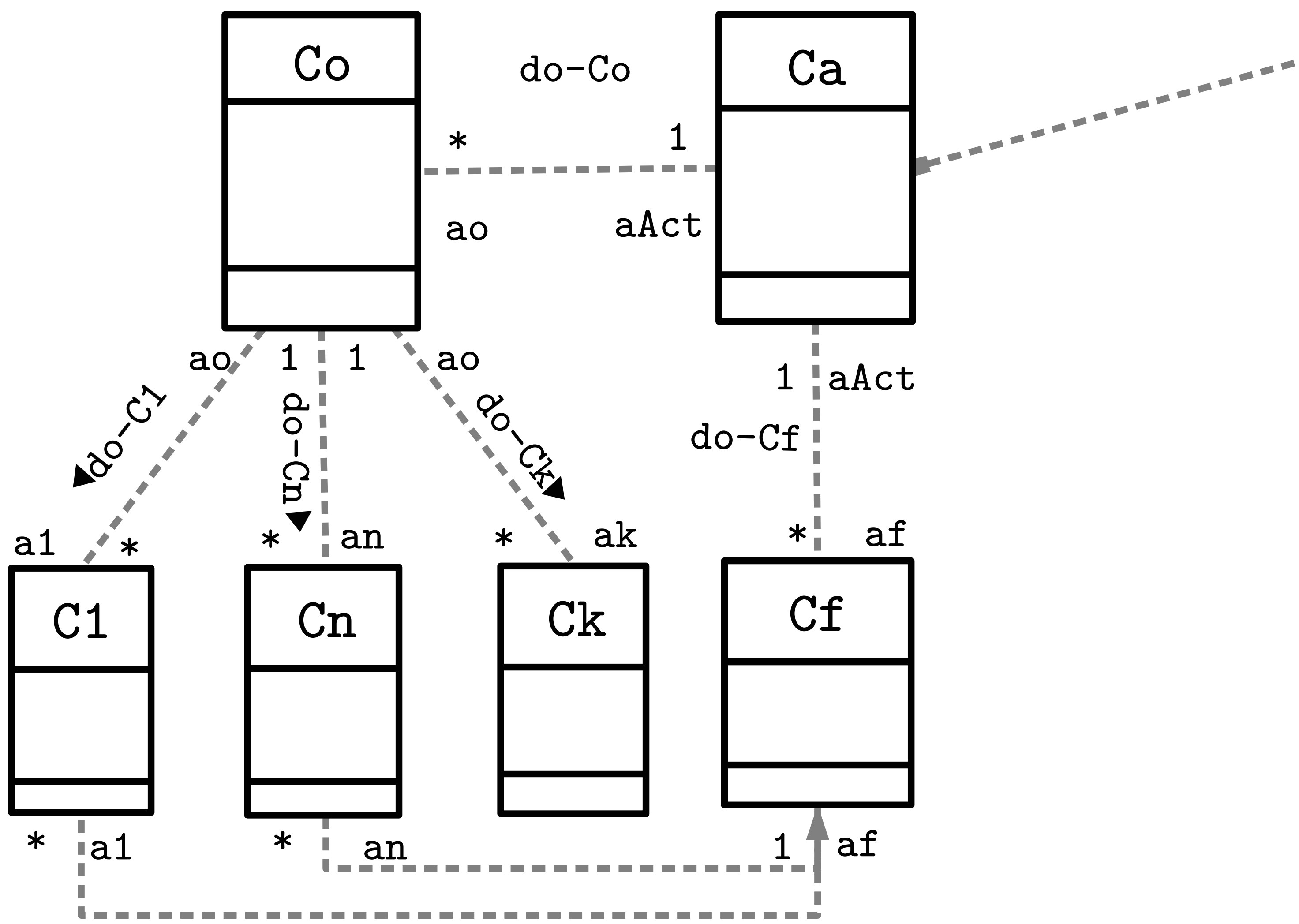
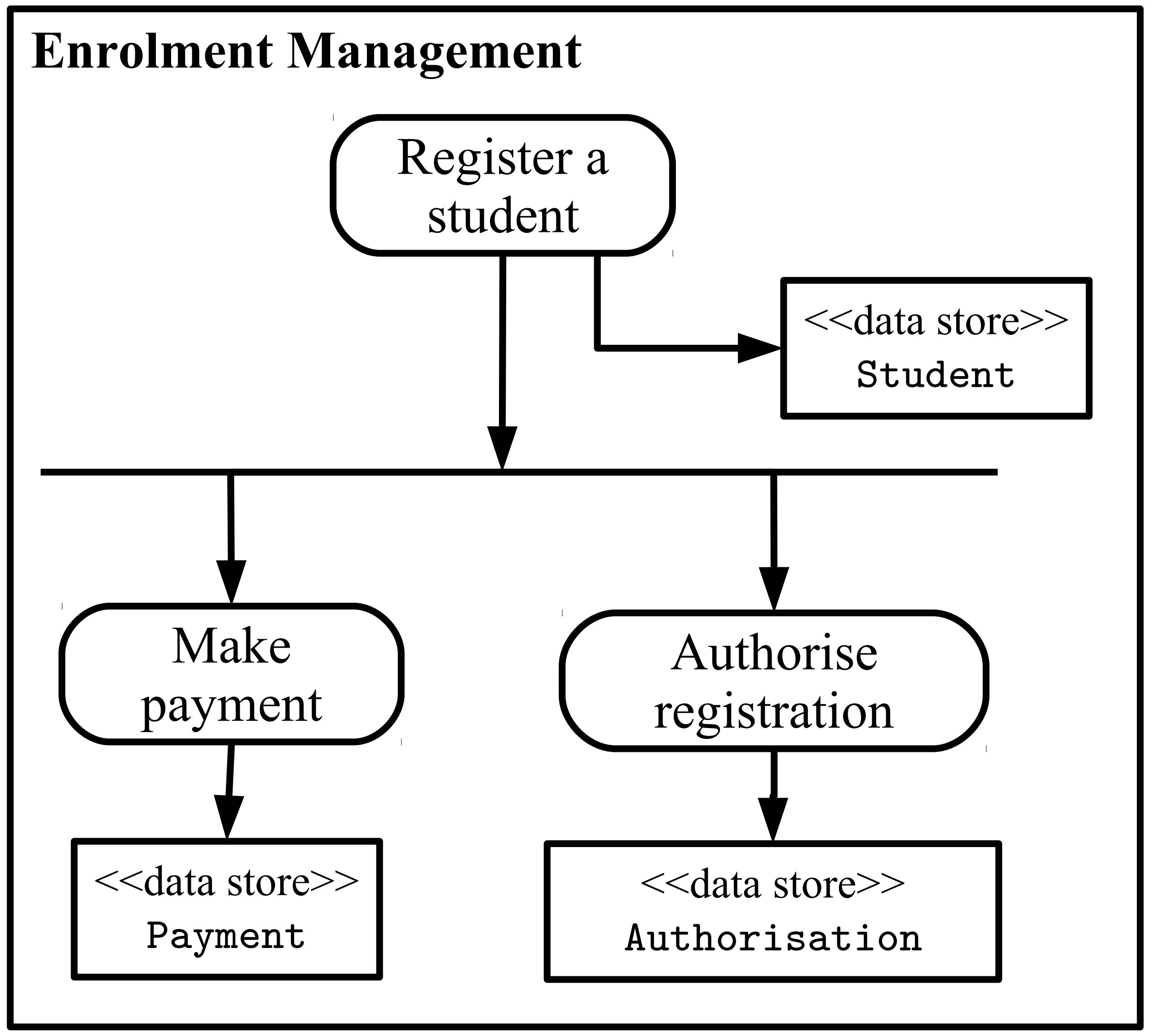
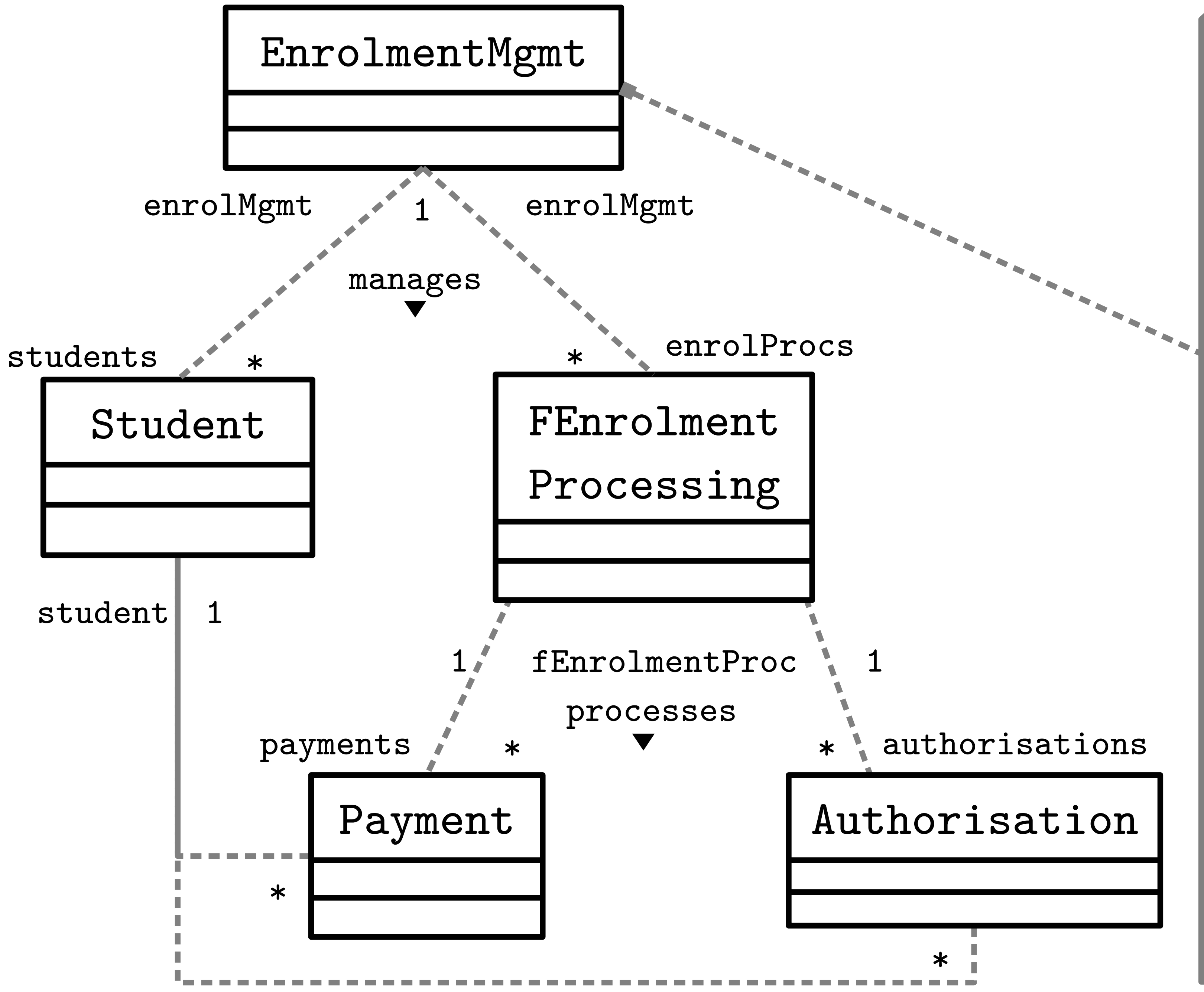
forked pattern
(if c_k is present then $n \geq 1$
else $n \geq 2$)





```
AGraph { nodes=[
  ANode { refCls=Cf,serviceCls=DataController
  outClses=[Co]
  actSeq=[
    MAct{actName=newObject,pstStates=[Created]} ]
  init=true}
  ANode { refCls=Co,serviceCls=DataController
  nodeType=Fork, outClses=[C1,Cn,Ck]}
  ANode { refCls=C1,serviceCls=DataController
  actSeq=[
    MAct{actName=newObject,pstStates=[NewObject]},
    MAct{actName=setDataFieldValues,fieldNames=["af"]
           pstStates=[Created]} ]}
  ANode { refCls=Cn,serviceCls=DataController
  actSeq=[
    MAct{actName=newObject,pstStates=[NewObject]},
    MAct{actName=setDataFieldValues,fieldNames=["af"]
           pstStates=[Created]} ]}
  ANode { refCls=Ck, nodeType=k, outClses=k_out}
 ]}
```





```
AGraph { nodes=[
  ANode { refCls=Student,serviceCls=DataController
  outClses=[FEnrolmentProcessing]
  actSeq=[
    MAct{actName=newObject,pstStates=[Created]} ]
  init=true}
  ANode { refCls=FEnrolmentProcessing
  serviceCls=DataController
  nodeType=Fork, outClses=[Payment,Authorisation]}
  ANode { refCls=Payment,serviceCls=DataController
  actSeq=[
    MAct{actName=newObject,pstStates=[NewObject]},
    MAct{actName=setDataFieldValues
         fieldNames=["student"], pstStates=[Created]} ]}
  ANode { refCls=Authorisation,serviceCls=DataController
  actSeq=[
    MAct{actName=newObject,pstStates=[NewObject]},
    MAct{actName=setDataFieldValues
         fieldNames=["student"],pstStates=[Created]} ]}
 ]}
```