**aspect ResourceSearch depends on Allocatable, ZeroToManyAssociation, Map**

IResource
ICapability

*structural view*

**ICapability**
+ Set<IResource> findResources()

myCapability
1

**IResource**
+ IResource create
(.., ICapability myCapability, ...)

*Allocatable instantiation*
**IAllocatable → IResource**

*ZeroToMany instantiation*
**IData → ICapability**
**IAssociated → IResource**

**IRequest**
+ Request create()
+ Set<IResource> find()
+ add(ICapability cap, Integer num)
+ Set<ICapability> getCapabilities()
+ int getNumber(ICapability c)
+ delete()

*Map instantiation*
**IKey → ICapability**
**IValue → Integer**
**IData → Request**

*state view IResource*

**Pointcut**

create → **Existing**

*Binding*
**Existing → ***

**Advice**

create → **Existing**

*Allocatable instantiation*
**IFree → Existing**
**IBusy → Existing**

*state view ICapability*

**Pointcut**

**Any**

**Advice**

**Any** findResources

*ZeroToManyAssociation instantiation*
**AddAllowed → Any**
**RemoveAllowed → Any**
**Any → Any**

*state view Request*

create → **Existing** delete

find

*Map instantation*
**AddAllowed → Existing**
**Any → Existing**

*message view findResources is ZeroToManyAssociation.getAssociated*

*message view initializeAssociation* instantiates ZeroToManyAssociation.add

**Pointcut**

**caller: Caller**
create(.., ICapability c, ..)
**new: IResource**

**Advice**

**caller: Caller**
create(.., ICapability c, ..)
**new: IResource**
add(new)
**c: ICapability**

*Binding*
**caller → *, Caller → *, new → ***

*message view cleanup* depends on ZeroToManyAssociation.remove

**Pointcut**

**caller: Caller**
delete(..)
**target: IResource**

*Binding*
**caller → ***
**Caller → ***
**target → ***

**Advice**

**caller: Caller**
delete(..)
**target: IResource**
remove(target)
**myCapability: ICapablity**

*message view find instantiates getAssociated, getValue, getKeys, isAllocated, insert*

**Pointcut**

**caller: Caller**
solution := find()
**target: Request**

*Binding*
**caller → ***
**Caller → ***
**target → ***

**Advice**

**caller: Caller**
solution := find()
**target: Request**
solution := create()
**solution: Set<IResource>**
keys := getKeys()

**loop** [k within keys]
i := getValue(key)
**i: Integer**
num := getValue()
**k: ICapability**
counter := 0
resSet := getAssociated()

**loop** [r within resSet and counter < num]
**r: IResource**
allocated := isAllocated()

**opt** [not allocated]
insert(r)
counter := ++

*message view add is Map.add*

*message view getCapabilities is Map.getKeys*

*message view getNumber instantiates Map.getValue*

**Pointcut**

**caller: Caller**
result := getNumber(c)
**target: Request**

*Binding*
**caller → ***
**Caller → ***
**target → ***

**Advice**

**caller: Caller**
result := getNumber(c)
**target: Request**
i := getValue(c)
**i: Integer**
result := getValue()