

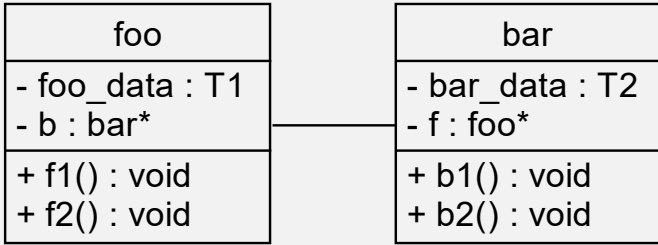


# USING ASSOCIATION

Simple to complex



# SIMPLE ASSOCIATION: GENERIC EXAMPLE



```
void foo::f2()
{
    b->b1();
}
```

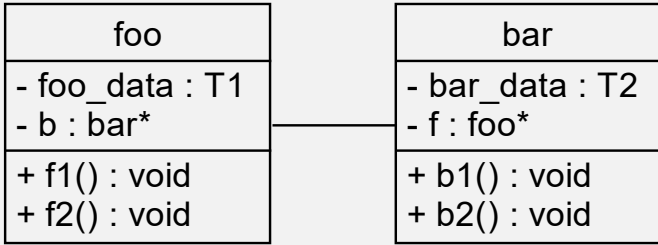
```
void bar::b2()
{
    f->f1();
}
```

```
void foo::f1()
{
    ...foo_data...
}
```

```
void bar::b1()
{
    ...bar_data...
}
```



# SIMPLE ASSOCIATION: GENERIC EXAMPLE

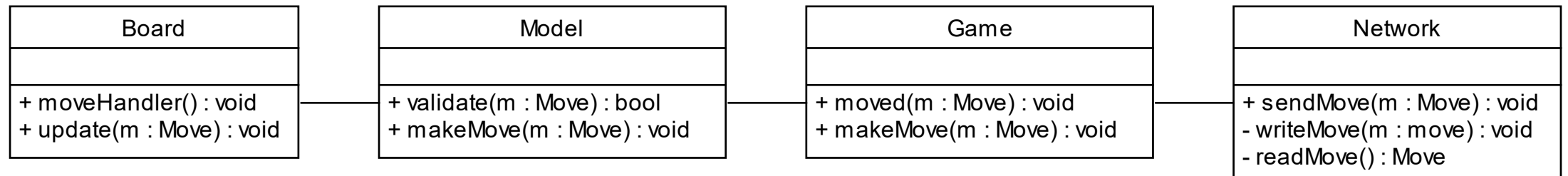


```
void foo::f2()
{
    b->b1();
}
```

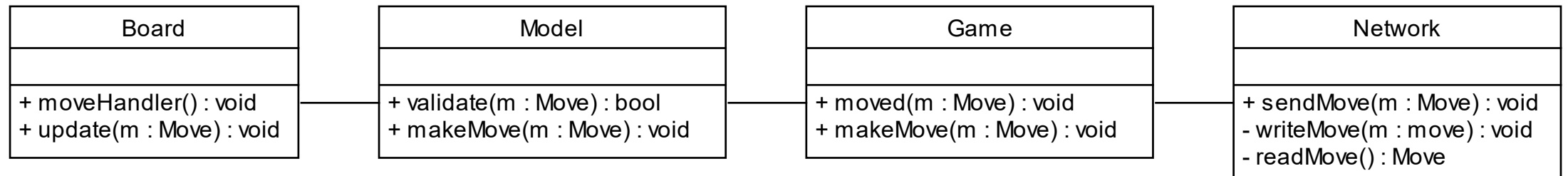
```
void bar::b2()
{
    f->f1();
}
```

```
void foo::f1()
{
    ...foo_data...
}
```

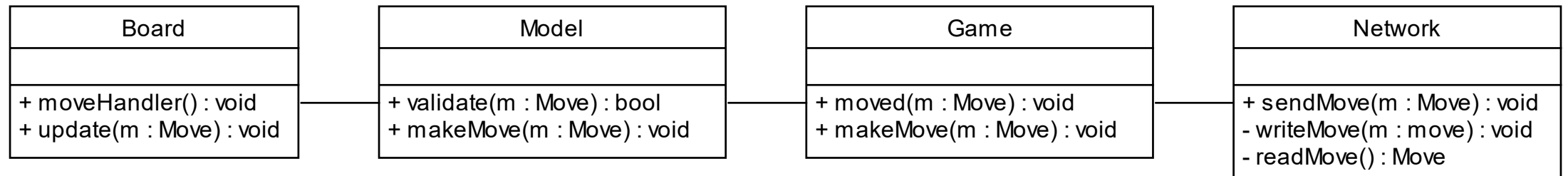
```
void bar::b1()
{
    ...bar_data...
}
```



COMPLEX BUT TYPICAL  
ASSOCIATION



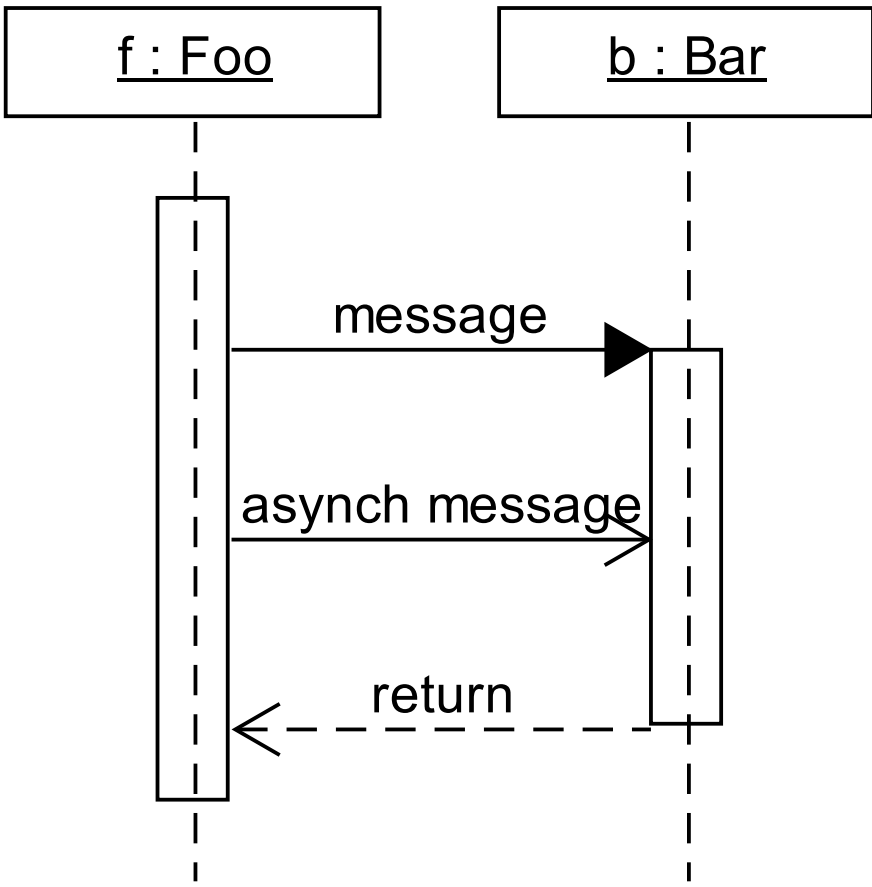
COMPLEX BUT TYPICAL  
ASSOCIATION

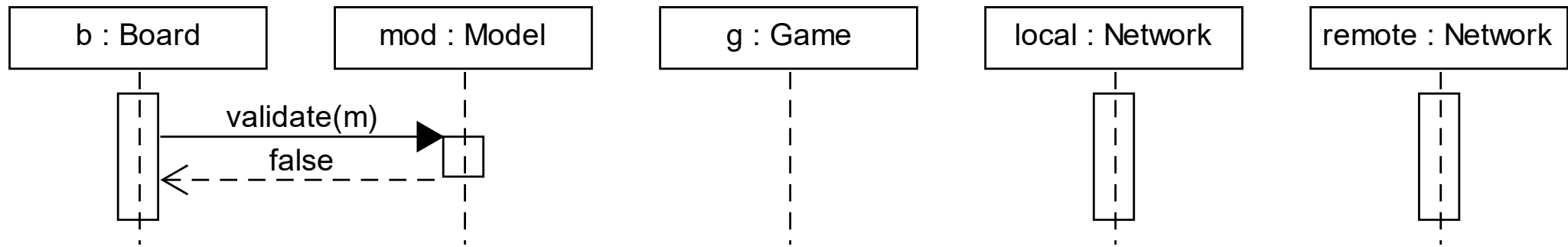


COMPLEX BUT TYPICAL  
ASSOCIATION



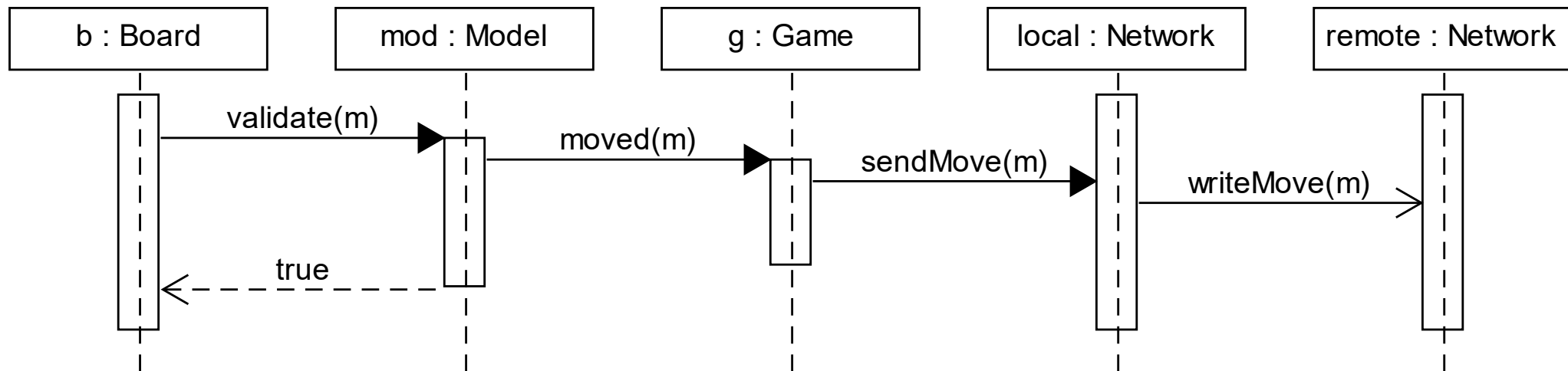
# UML SEQUENCE DIAGRAMS



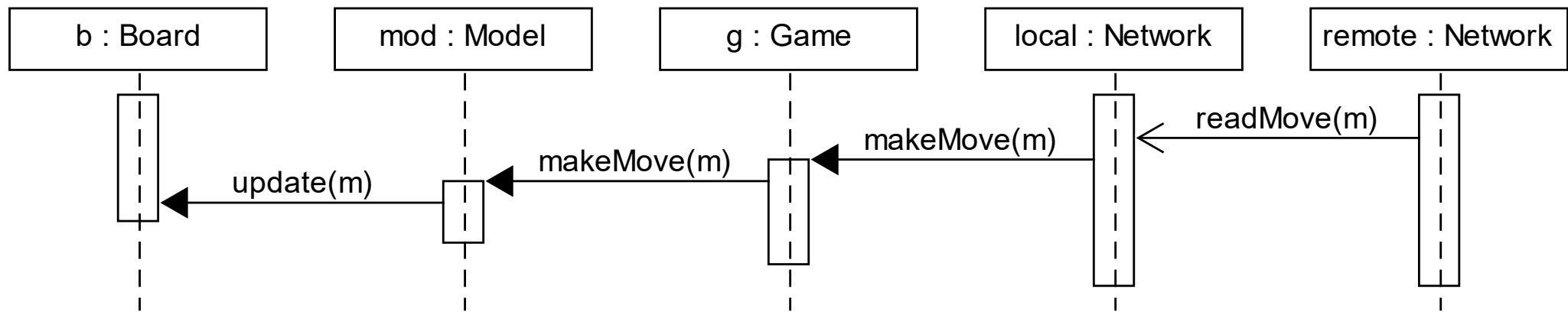


**SCENARIO I:  
MAKING AN ILLEGAL MOVE**





**SCENARIO 2:  
MAKING A LEGAL MOVE**



**SCENARIO 3:  
RECEIVING A MOVE**

## ASSOCIATION ERROR: INFINITE CYCLE

```
class project
{
    private:
        string    title;
        contractor* c;

    public:
        void display
        {
            cout << title << endl;
            c->display();
        }
};
```

```
class contractor
{
    private:
        string    name;
        project*  p;

    public:
        void display()
        {
            cout << name << endl;
            p->display();
        }
};
```