

# INGÉNIERIE DE LOGICIELLE

❖ **ENSEIGNANT:** Yann Thierry-Mieg

❖ **MEMBRES:**

- ✓ BENCHERIF Nabil
- ✓ BOUAOUD Malik
- ✓ CARET Romain
- ✓ EAR Sandrine
- ✓ PAREDES Lizeth

❖ **GROUPE:** Terminator

*Pour un logiciel qui termine vos livres*



# Sommaire

## 1. Rappel de la partie Analyse

### 1.1 Diagramme de Classe Métier

## 2. Conception Détaillée

### 2.1 Diagramme de Composant

#### 2.1.1. Diagramme de Composant : Livre

#### 2.1.2. Diagramme de Composant : Section/Enchaînement

### 2.2. Diagramme de Séquence

#### 2.2.1 Diagramme de Séquence : Gestion Livre

#### 2.2.2 Diagramme de Séquence : Section/Enchaînement

## 3 Diagramme de Structure Interne

## 4. Test d'intégration & Tests Unitaires

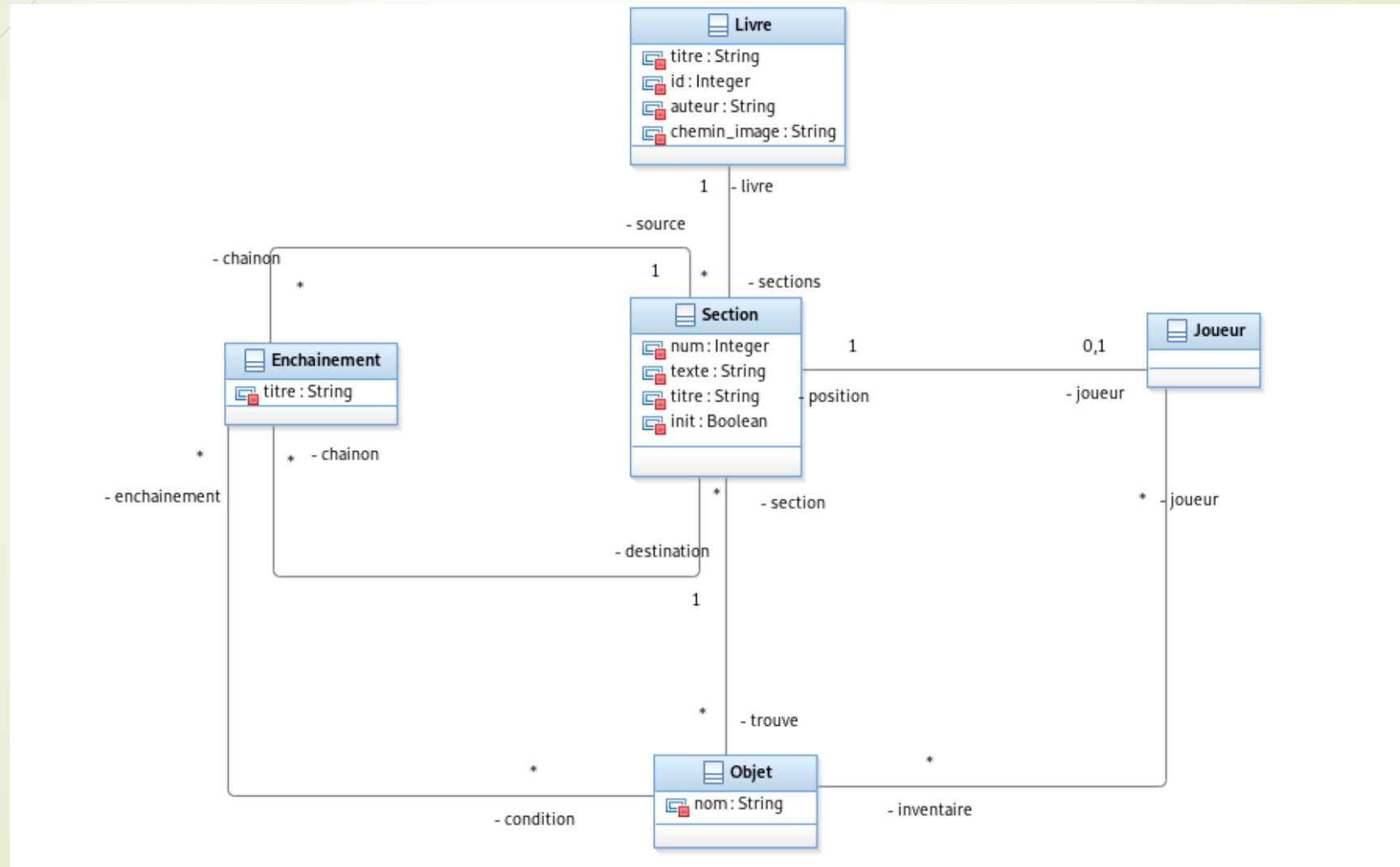
### 4.1. Diagramme de Composant : Test

### 4.2. Diagramme de Composant: Factory

### 4.2. Tests Unitaires

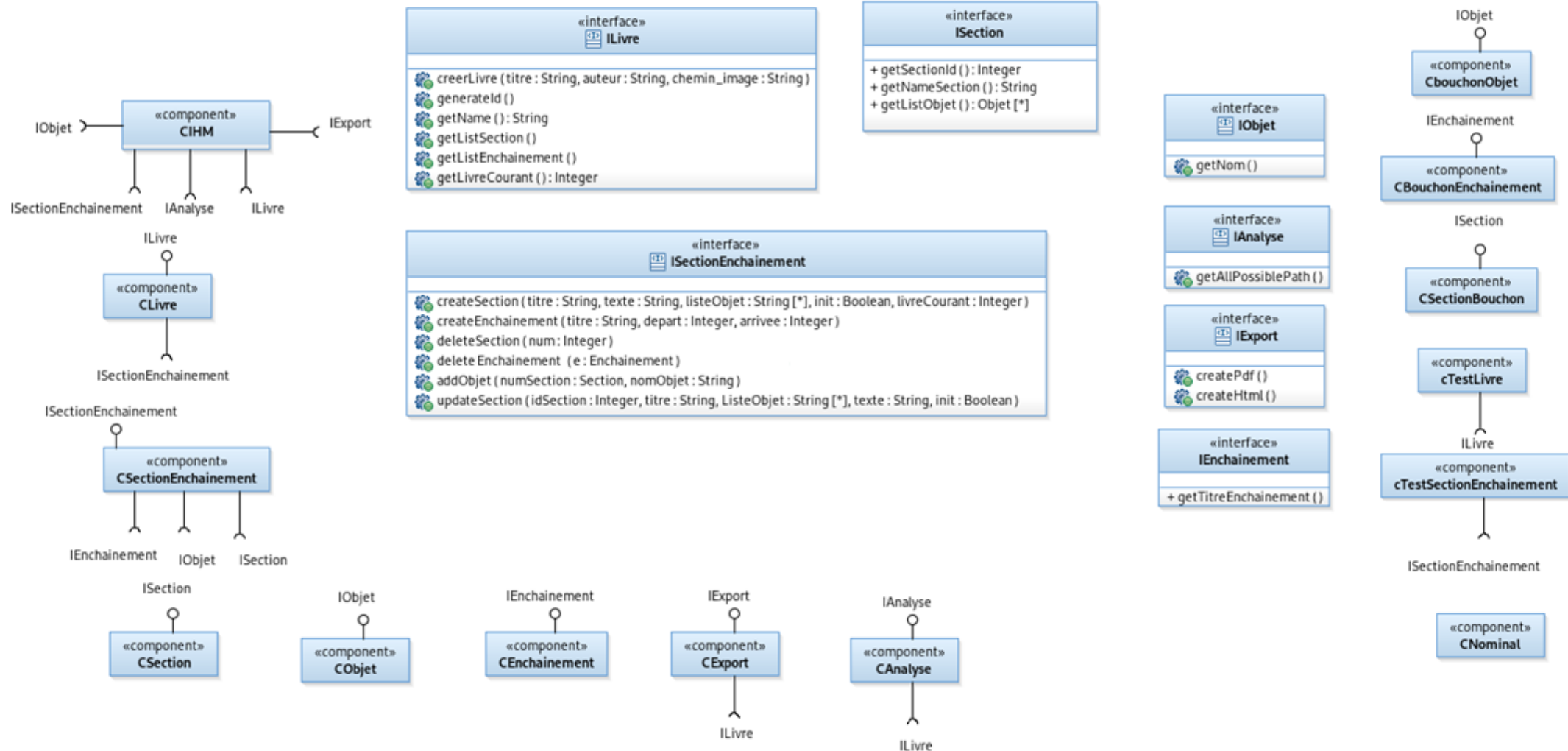
# 1. Rappel de la partie Analyse

## 1.1. Diagramme de Classe Métier



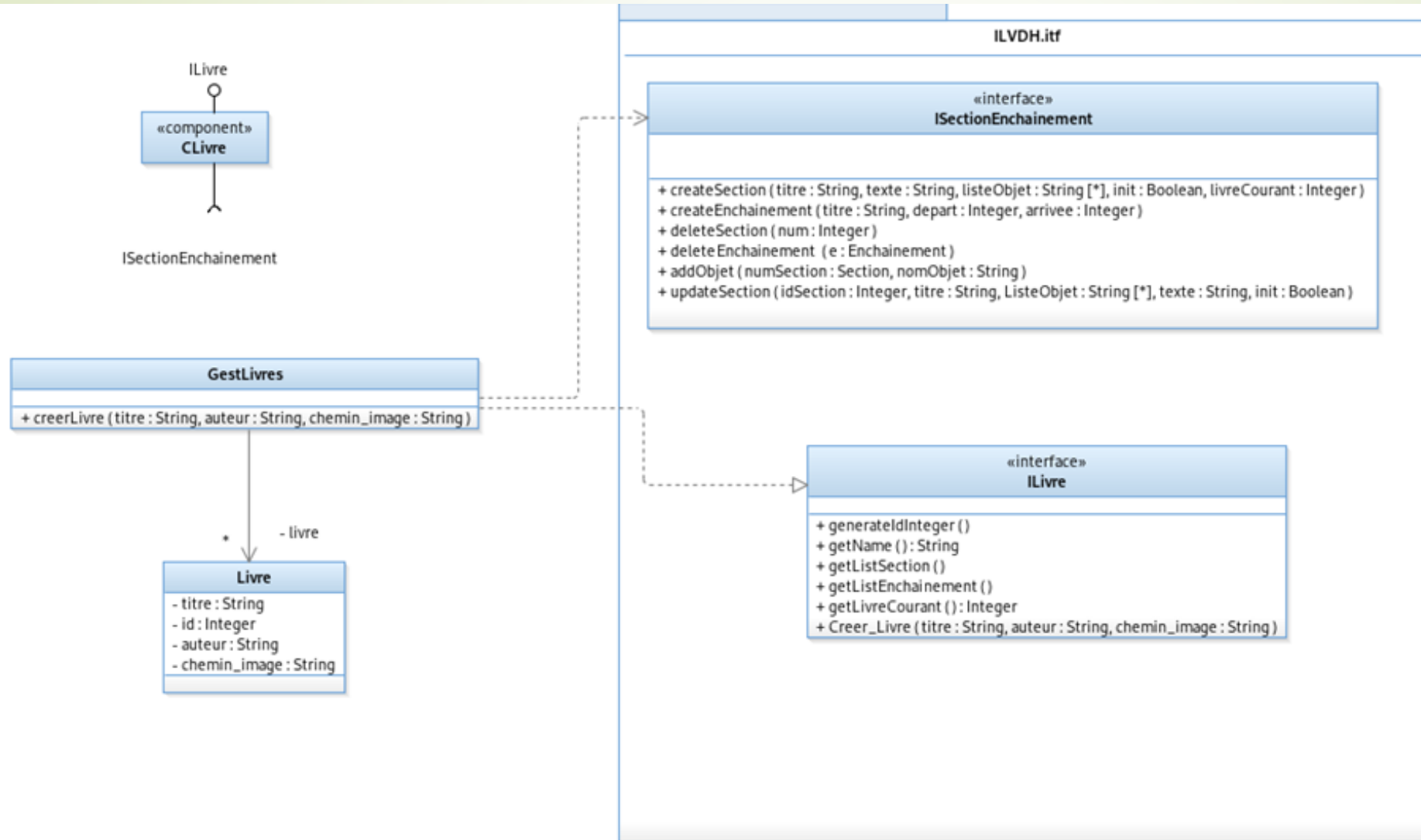
## 2.0. Conception Détaillée

### 2.1 Diagramme de Composant



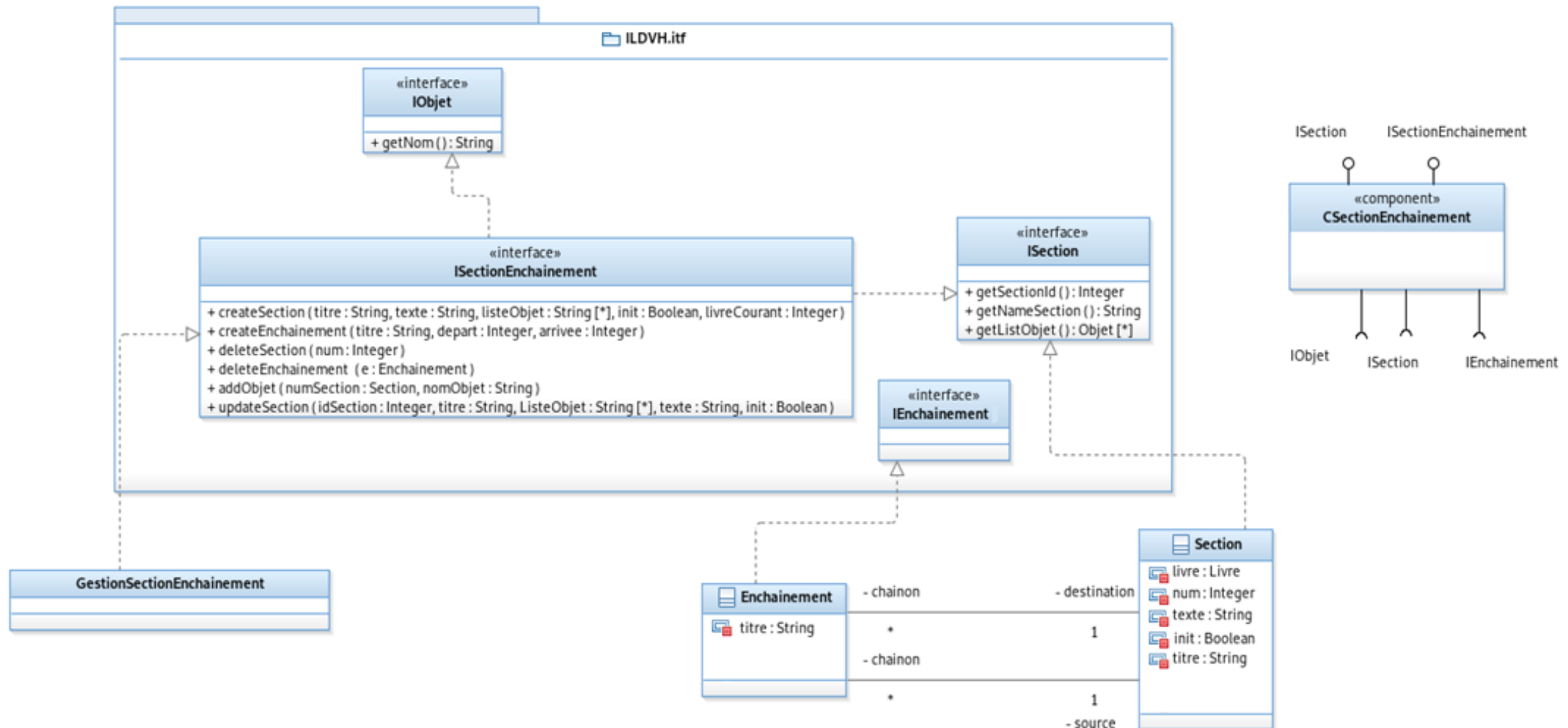
## 2.1. Diagramme de Composant

### 2.1.1. Diagramme de Composant : Livre



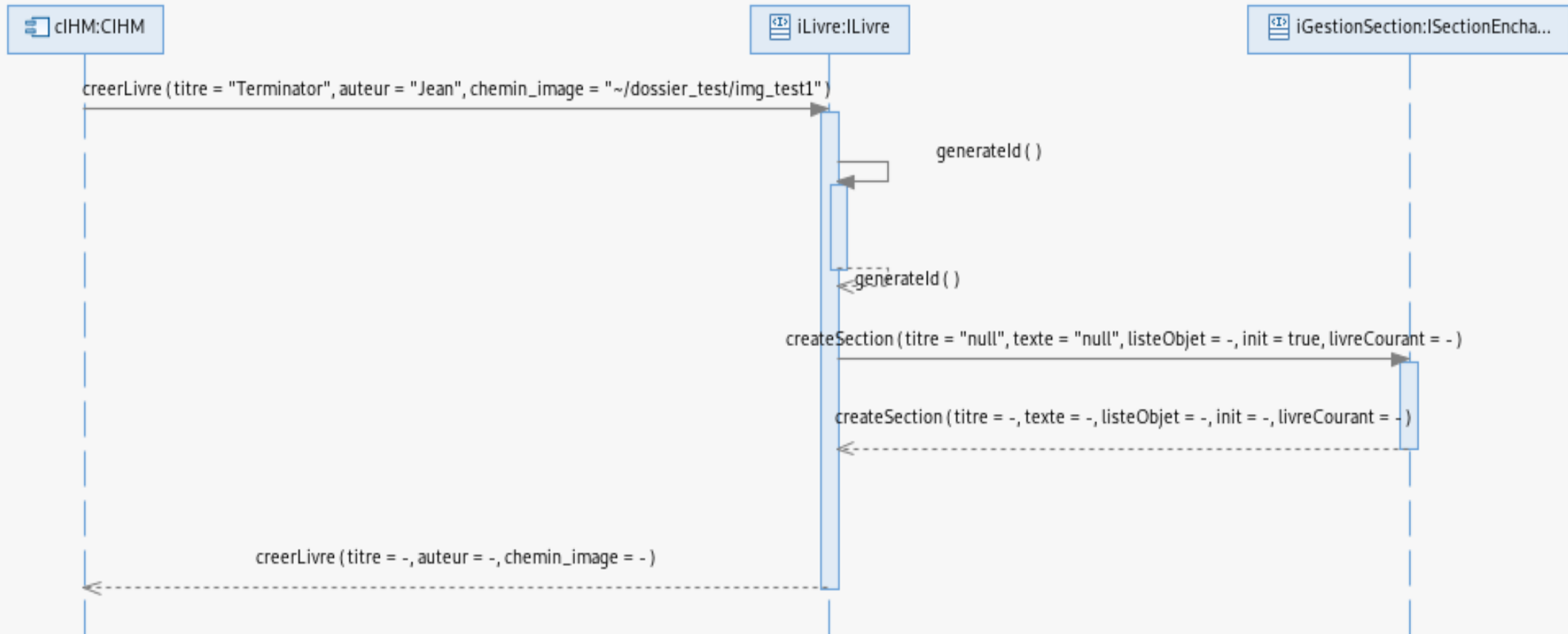
## 2.1 Diagramme de Composant

### 2.1.2. Diagramme de Composant : Section/Enchaînement



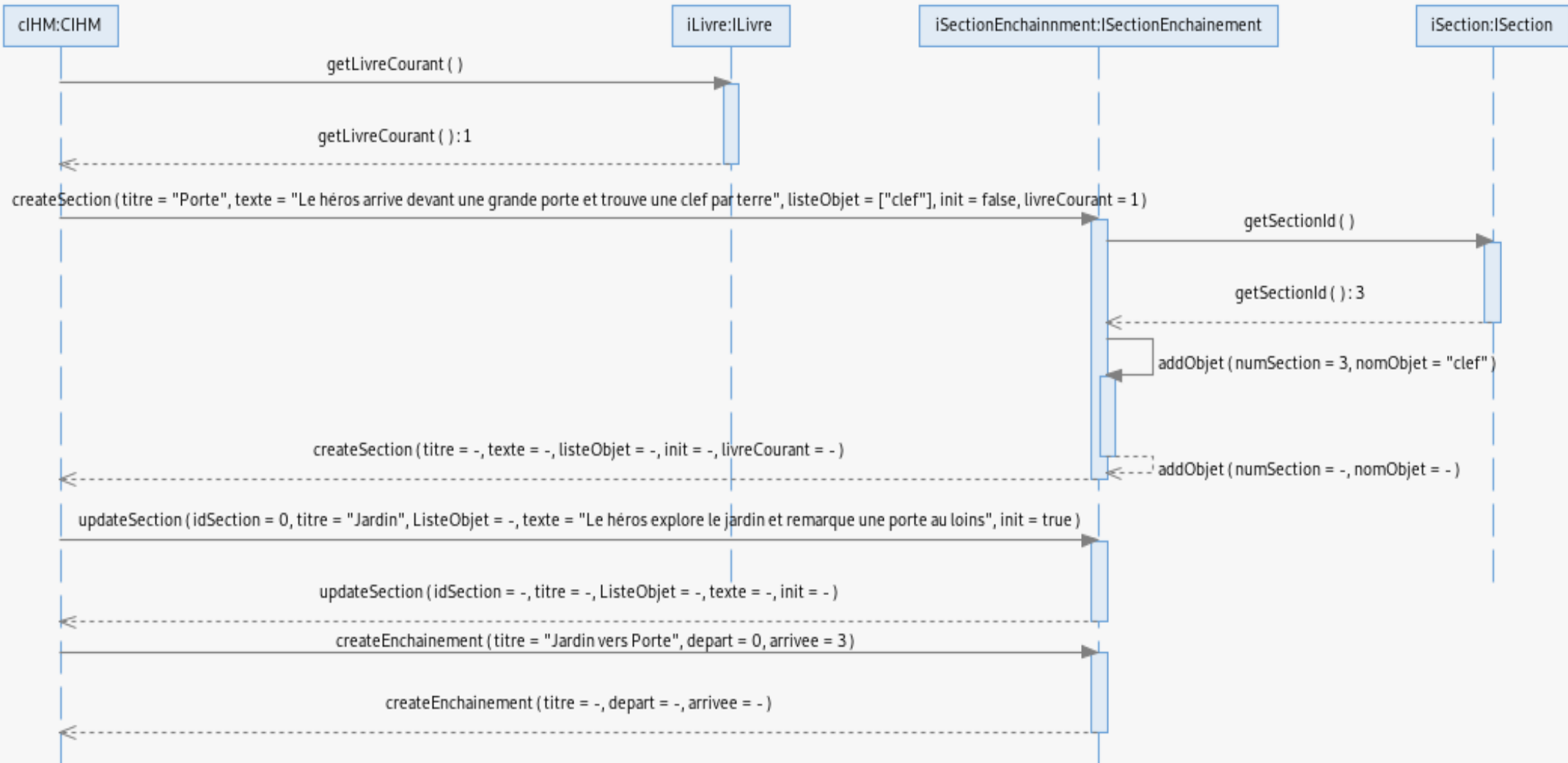
## 2.2. Diagramme de Séquence

### 2.2.1 Diagramme de Séquence : Gestion Livre



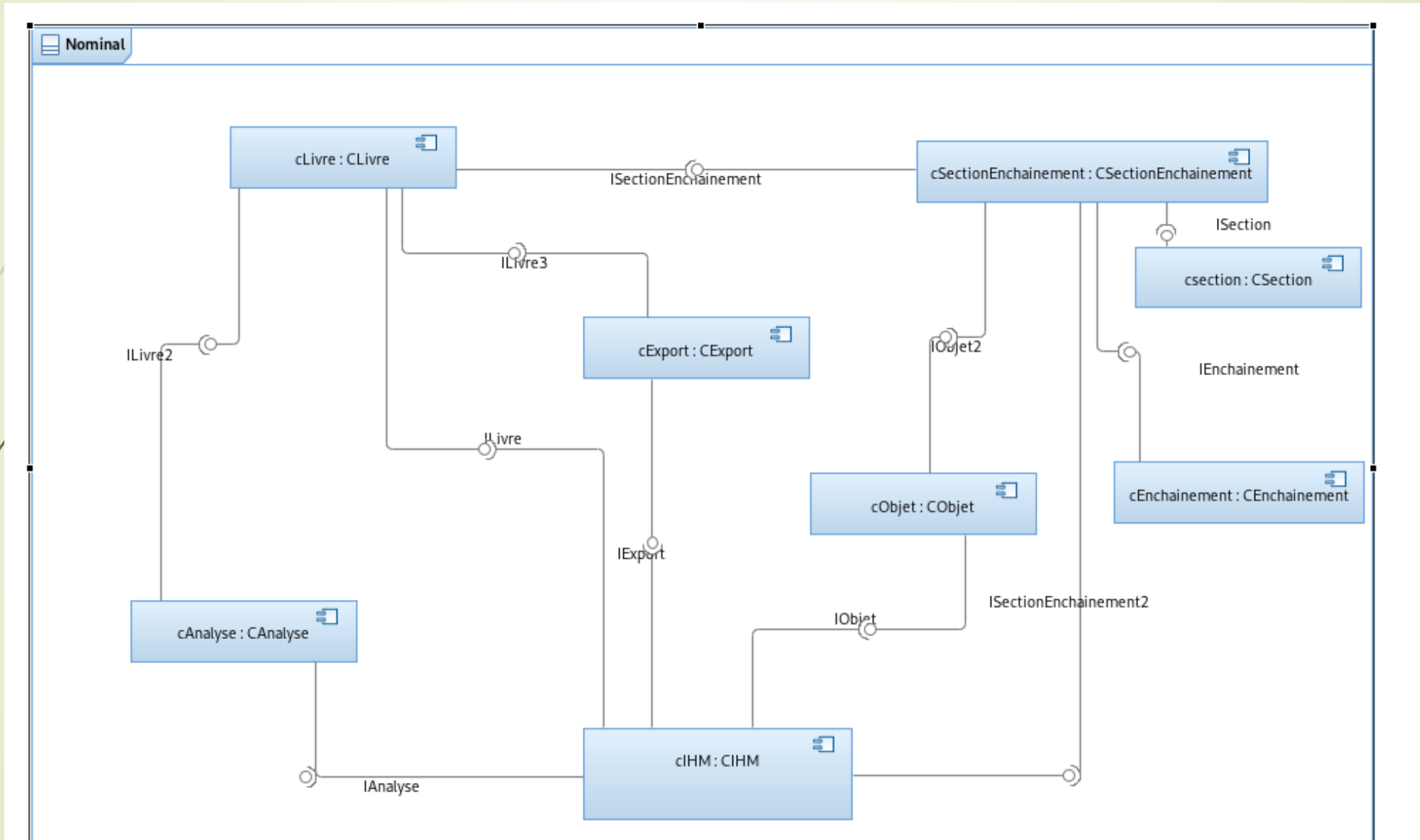
## 2.2. Diagramme de Séquence

### 2.2.2 Diagramme de Séquence : Section/Enchaînement



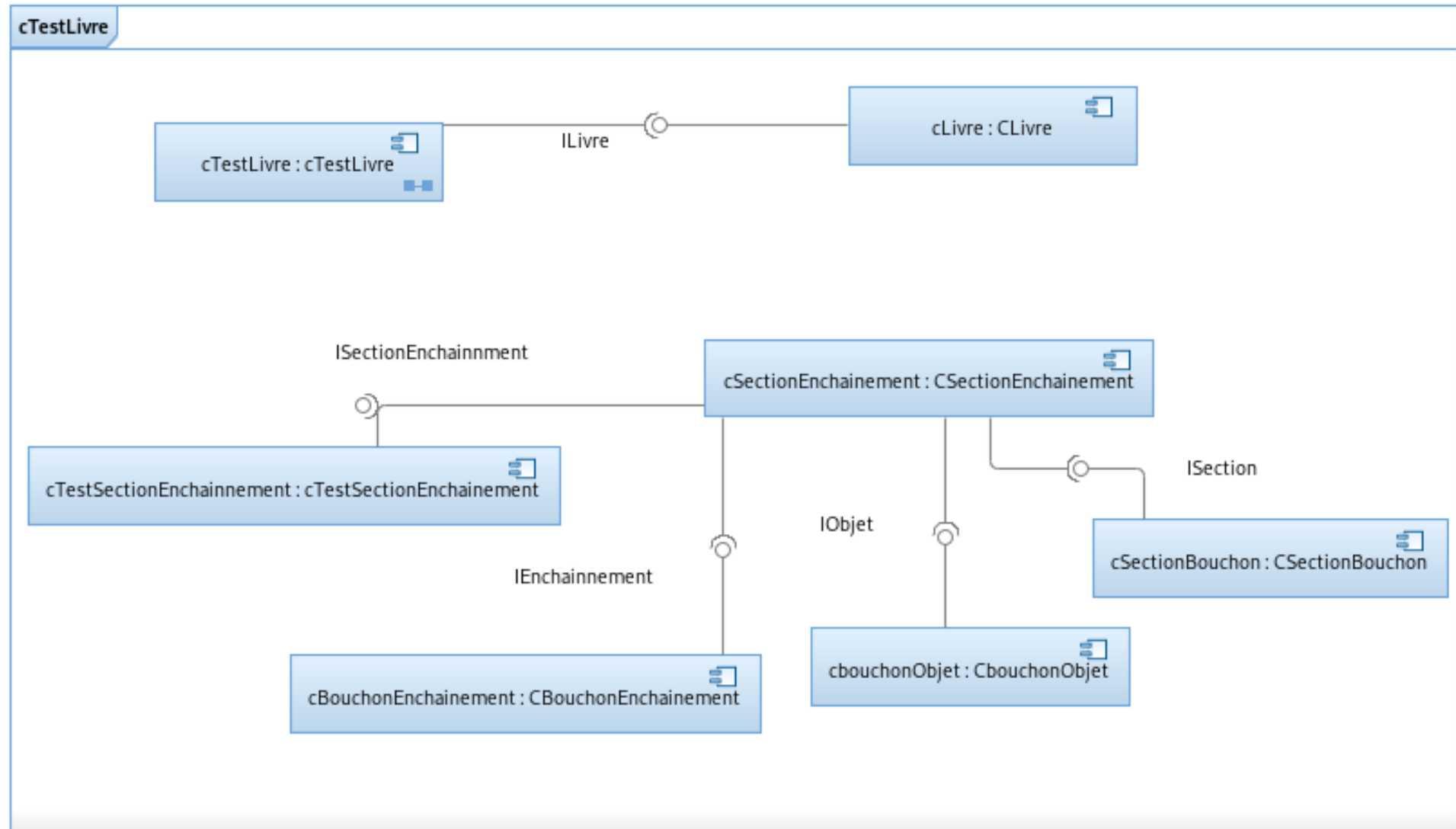


### 3. Diagramme de Structure Interne



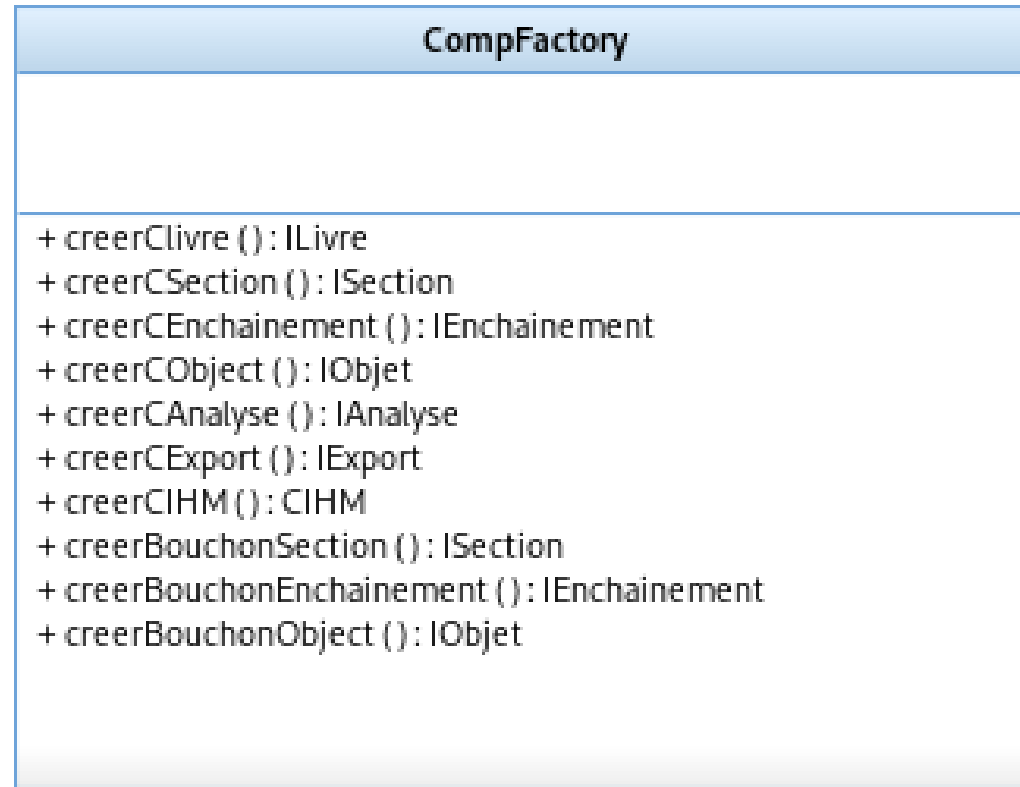
## 4. Test d'intégration & Tests Unitaires

### 4.1 Diagramme de Composant : Test



## 4. Test d'intégration & Tests Unitaires

### 4.2. Diagramme de Composant : Factory



## 4. Test d'intégration & Tests Unitaires

### 4.3 Tests Unitaires - 1er

```
@Test
    public void testLivre(){

        CompFactory factory=new CompFactory();

        ILivre l = facotry.creerLivre("Terminator","Jean","/path/img");
        this.assertEquals("Terminator",l.getName());
        this.assertEquals("Jean",l.getAuteur());
        this.assertEquals("/path/img",l.getPath());
        this.assertTrue(1==l.getListSection().size());

    }
```

## 4.4 Tests Unitaires - 2nd

```
@Test
public void testSectionEnchainement(){
    CompFactory factory=new CompFactory();
    ILivre l = facotry.creerLivre("Terminator","Jean","/path/img");
    ISectionEnchainement s1=factory.createSection("jardin","Le heros se trouve dans un
jardin",null,false,l.getIdLivre());
    ISectionEnchainement s2=factory.createSection("porte","Le heros se trouve devant
une porte",null,false,l.getIdLivre());
    ISectionEnchainement e=factory.createEnchainement("jardin vers porte",s1,s2);

    this.assertEquals(s1.getNameSection(),"jardin");
    this.assertEquals(s1.getNameSection(),"porte");
    this.assertEquals(e.getTitreEnchainement(),"jardin vers porte");

}
```



**Merci pour votre attention**