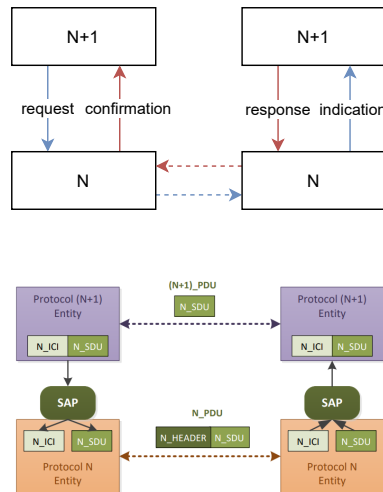


1 Introduction

1.1 OSI



Un SAPI permet de choisir quel protocole utiliser (si il y a plusieurs protocoles)

2 Protocole

2.1 Primitives de service

- request : couche N+1
- indication : couche N
- response : couche N+1
- confirm : couche N

3 C++

3.1 friend

```
class A {
private:
    // La classe B peut accéder aux méthodes
    privées de A
    friend class B;
    // La fonction calc de C peut accéder aux
    méthodes privées de A
    friend int C::calc(int x);
}
```

```
// La fonction main peut accéder aux mé
thodes privées de A (à éviter)
friend int main();
}
```

3.2 Polymorphisme

3.2.1 static binding

```
class A {
public:
    void display() { cout << "A" << endl;};
};
class B : public A {
public:
    void display() {cout << "B" << endl;};
};

int main() {
    A a;
    B b;
    A* p;
    p = &a;
    p->display(); -> "A"
    p = &b;
    p->display(); -> "A"
}
```

3.2.2 Dynamic binding

```
class A {
public:
    virtual void display() { cout << "A" <<
        endl;};
};
class B : public A {
public:
    virtual void display() {cout << "B" << endl
        };};

int main() {
    A a;
    B b;
    A* p;
    p = &a;
    p->display(); -> "A"
    p = &b;
    p->display(); -> "B"
}
```

```
}
```

3.2.3 Interfaces

```
class IVehicle {
public:
    virtual void drive() = 0;
};
class Car : public IVehicle {
public:
    virtual void drive() { cout << "car drives"
        << endl;};
};
class Rocket : public IVehicle {
public:
    virtual void drive() { cout << "rocket
        flies" << endl;};
};

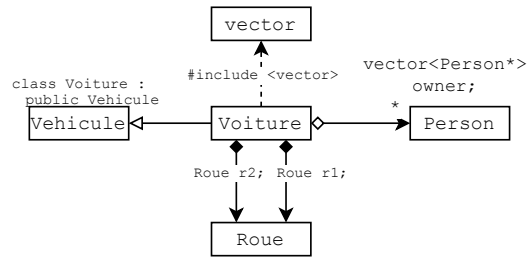
int main() {
    IVehicle* v1 = new Rocket();
    IVehicle* v2 = new Car();
    v1->drive(); // car
    v2->drive(); // rocket
    delete v1;
    delete v2;
};
```

3.3 Classes génériques

```
vector<T> vInt(5);
```

3.4 UML

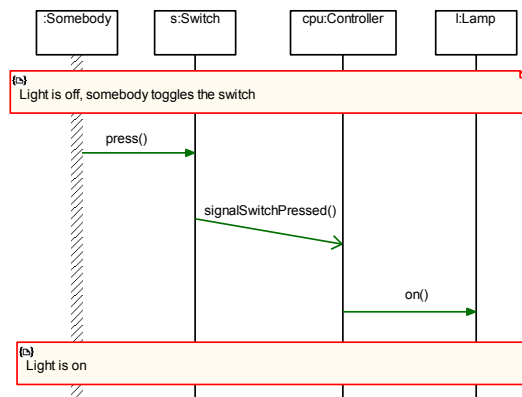
association	Appelle entre deux classes
héritage	Spécification de classe (est un)
composition	objet dont la durée de vie est liée à la classe (variable)
agrégation	objet dont la durée de vie n'est pas liée à la classe (pointeur)
include	Implémentation d'une autre classe



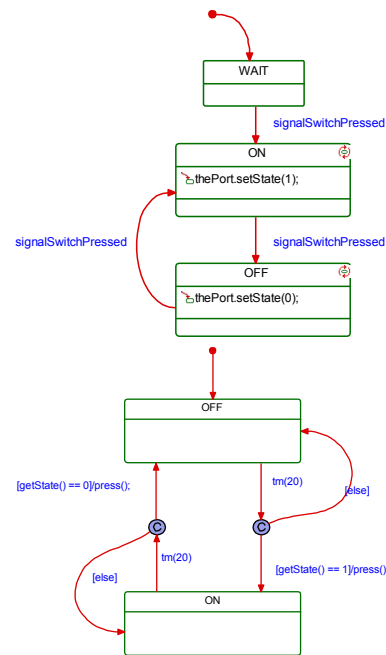
3.4.1 Diagrammes de classe

ClassName
<pre> - privateAttribute : type # protectedAttribute: type + publicAttribute: type - privateMethod(param1 : type, ... paramN : type) : type # protectedMethod(param1 : type, ... paramN : type) : type + publicMethod(param1 : type, ... paramN : type) : type </pre>

3.4.2 Diagrammes de séquences



3.4.3 Diagrammes d'états



3.5 Singleton

```

class Singleton {
public:
    static Singleton& getInstance() {
        static Singleton instance;
        return instance;
    }
private:
    // Private constructor and destructor
    Singleton() {};
    Singleton(const Singleton&) {};
    void operator=(const Singleton&) {};
};

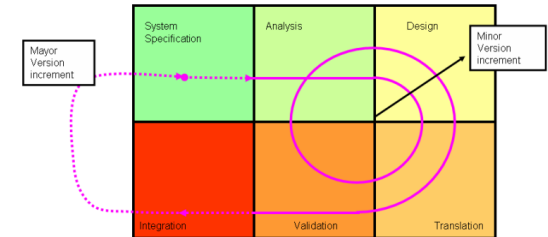
int main() {
    Singleton::getInstance().doSomething();
}

```

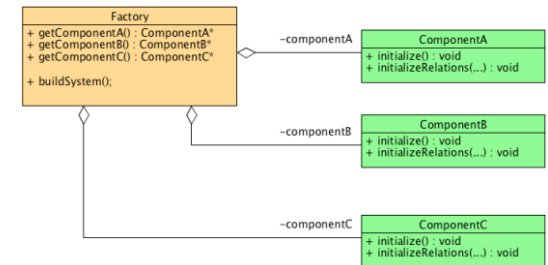
4 Patterns

4.1 Process

4.1.1 6Q

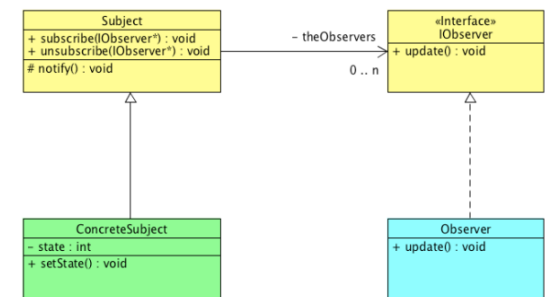


4.2 Factory

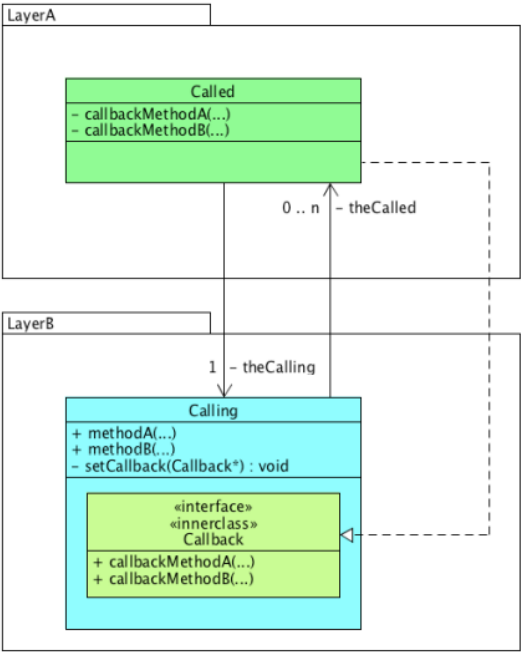
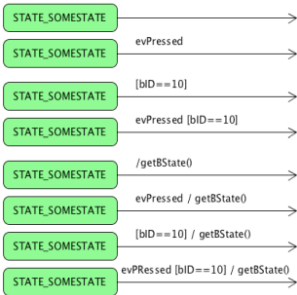
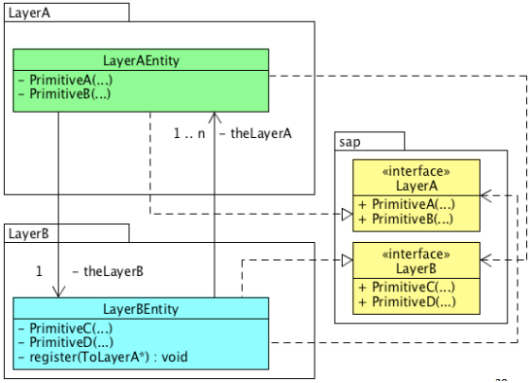


1. create
2. initialize
3. initializeRelations quand tous les create et initialize sont faits

4.3 Observateur

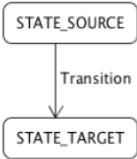


4.4 Interface



5 XF et Machines d'états

- 1. États
 - (a) Action sur entrée
 - (b) Action continue
 - (c) Action sur sortie
- 2. Transitions
 - (a) déclenchée par un événement
 - (b) Condition
 - (c) Action

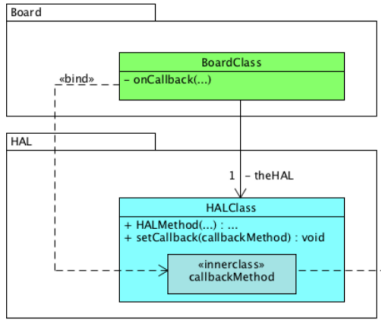


5.1 XF

Atomique : Pas d'interruptions

6 HAL

- 1. Portabilité
- 2. Réutilisable



7 DeseNET

N	Nom	unité	description	exemples
7	Application	Données	Utilité pour l'utilisateur (transfert de fichiers, vidéos, etc...)	Web, FTP, IMAP, LDAP, HTTP, SMB
6	Présentation	Données	Formats, mises en formes, cryptage, login	JSON, ASCII, HTML, Unicode
5	Session	Données	Gestion de l'activité	RPC, NetBios
4	Transport	Segments, streams	sous-adressage, communication entre deux processus	TCP, UDP
3	Réseau	Packets	Transport des données dans un réseau maillé	IPv4/IPv6, ARP
2	Liaison	Trame	Adressage local, gestion des erreurs, etc...	Ethernet, CAN,
1	Physique	Bit	Signaux électriques	Wi-Fi, Câble, 1000BASE-T, USB

Les couches 1-4 permettent de transférer les données. Les couches 5-7 sont liées à l'utilisation qu'on fait des données.