

1. Utilisation Enterprise-architect (outils UML2)

Enterprise Architect (...v15,...) est un **outils UML très complet** mais par certains cotés assez **complexe** (beaucoup de paramètres possibles , pas toujours intuitifs)

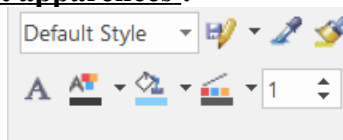


URL du site officiel de "Enterprise Architect" : <https://sparxsystems.com/>

documentation officielle ---> <https://sparxsystems.com/resources/user-guides/15.0>

Quelques généralités :

- Masquer (ou pas) le bandeau principal (gros menu)
--> click droit (sur bandeau/toolbar) et "**minimize the ribbon**"
- Faire apparaître certaines fenêtres (propriétés,notes,toolbox) (pour paramètres, ...) :
--> **Start / Design / Properties** (ou **Toolbox** ou **Notes** ou)
- Générer une image à partir du diagramme courant :
--> **Pubish/ save / image / save to file/** PNG ou SVG ou JPEG
- Contrôler les éléments affichés et masqués sur un diagramme :
--> **click droit** (dans fond du diagramme) et **properties .../ diagram** / cocher ou décocher "**show namespace**" ou autres .
et **properties .../elements/show compartments/** cocher ou décocher "**package contents**" ou autres .
- Changer couleur de fond et apparences :



--> dans menu "**Layout**",

1.1. Organisation des modèles (facultatif)

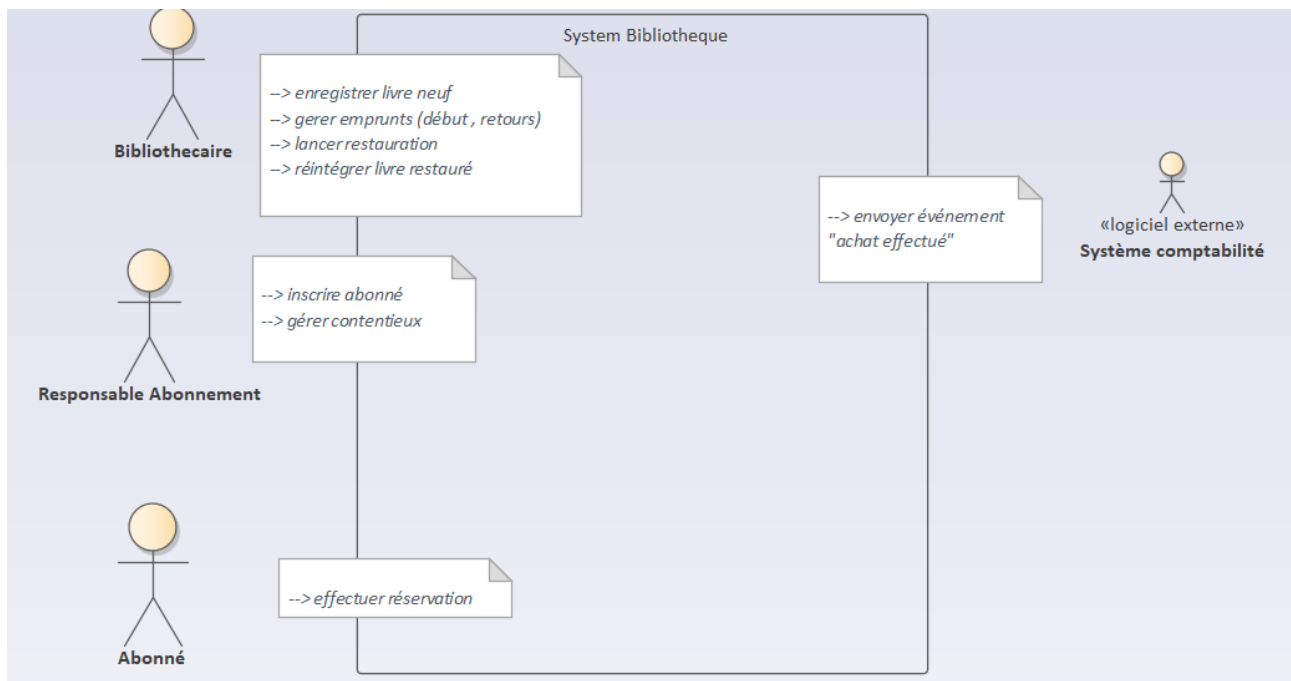
Se placer sur la racine "**Model**" et activer le menu contextuel "**Add a model using a wizard ...**",

- choisir ensuite la catégorie de sous modèle (ex : class ou useCase) puis une sous catégorie (souvent starter)
- Souvent besoin de renommer ensuite le sous modèle créé (en supprimant le qualificatif "starter") .

1.2. Edition d'un diagramme de cas d'utilisation préliminaire (diag de contexte)

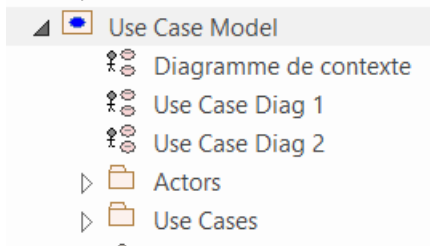
NB : Un diagramme de contexte est un diagramme de "use cases" préliminaire simplifié où l'on place volontairement des notes/commentaires (avec interactions) à la place des cas d'utilisation pour gagner en compacité/espace et faire en sorte que l'on puisse garder une vue d'ensemble imprimable sur une page au format A4 .

- > Add a model using a wizard .../ use cases / starter ...
- > renommer les acteurs A et B
- > supprimer les uses cases exemples A et B du modèle
- > placer des notes à la place avec des listes d'interactions.



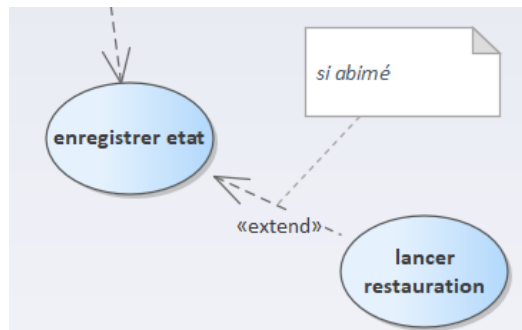
1.3. Edition d'un diagramme de cas d'utilisation

Se placer au niveau de "use case model" (préalablement créé) dans l'arborescence du modèle.
activer le menu contextuel "add diagram ..." et choisir "Uml behavior / use cases" .



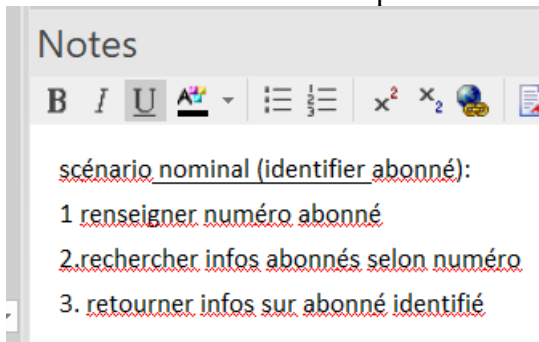
Manipulations moyennement intuitives :

- Les associations (entre acteur et U.C.) sont par défaut "non orientées".
On peut éventuellement modifier la propriété **direction** (en remplaçant "unspecified" par "**source--> destination**") pour placer certaines **flèches de navigabilité** .
Bien penser à sauvegarder le nouveau paramétrage .
- Pour relier un relation "<<extend>>" à une note/commentaire , le plus simple consiste à partir du lien "extends" activer contextuellement (via click droit) la flèche et de relâcher la souris sur le commentaire .



1.4. Edition rapide d'un scénario (pour U.C.)

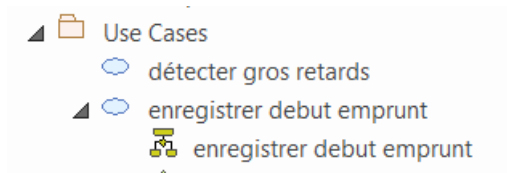
Sélectionner un "Use Case" puis saisir le scénario dans la partie "Notes" :



1.5. Ajout d'un diagramme d'activité (en tant que détails d'un U.C.)

Sélectionner un "use case" dans l'arborescence puis activer le menu contextuel

"Add / Add Diagram ... / Activity "



Le diagramme d'activité ainsi rattaché au "use case" sera ainsi clairement vu comme un processus métier permettant d'atteindre l'objectif utilisateur lié au "use case" .

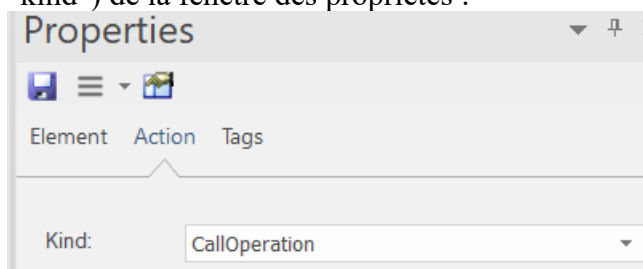
1.6. Edition d'un diagramme d'activité

1. placer (si besoin seulement) les "couloirs/partitions"
2. placer assez tôt un "début" et une (ou plusieurs) fin(s)
3. placer certaines actions , décisions, ...
4. placer seulement ensuite les "control flow"
5. les conditions (alias gardiens) doivent être précisées via la propriété **"gard :"** et devraient normalement apparaître entre des crochets []

paramétrage important pas intuitif :

modifier la propriété "gard" ne fonctionne pas bien dans la palette des propriétés principales. il faut modifier la valeur de **"gard"** dans la partie **"constraints"** (*autre onglet*) !!!
Ceci soit après un double click (ou un "open") sur un "control flow" soit via un paramétrage sauvegardé ...

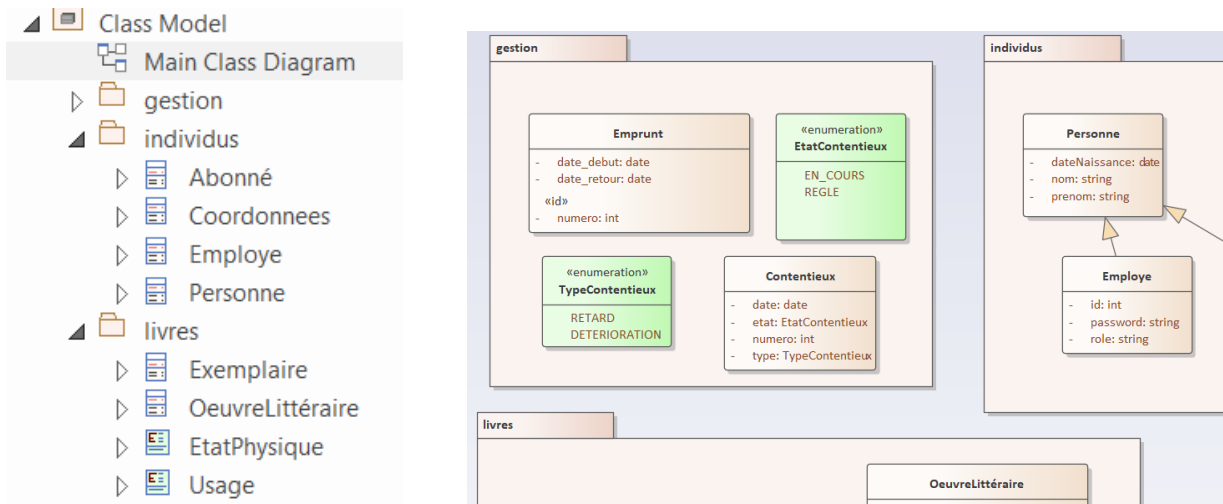
NB : en début de modélisation (expression de besoins) , on est encore assez loin des détails de conception et l'on peut utiliser essentiellement l'**action de type "atomic" (la plus basique)** .
On pourra (si besoin seulement) changer le type d'action par la suite via l'onglet action (et partie "kind") de la fenêtre des propriétés :



1.7. Organisation des diagrammes de packages & classes

- Commencer par ajouter un (sous)-modèle de classes via **"Add a model using a wizard ..."** et en choisissant "classes/ starter ..." .
- Renommer les choses (ex : ~~Starter Class Diagram~~ **Class Model** et ~~starter main~~ **main class diagram**)
- Supprimer les classes exemples "A" et "B" .
- En se plaçant dans l'arborescence au niveau du sous modèle qui vient d'être créé et renommé (ex : Class Model) , on peut ensuite activer le menu contextuel "Add package ..." pour ensuite choisir le nom du package à créer (idéalement en mode "package only") .

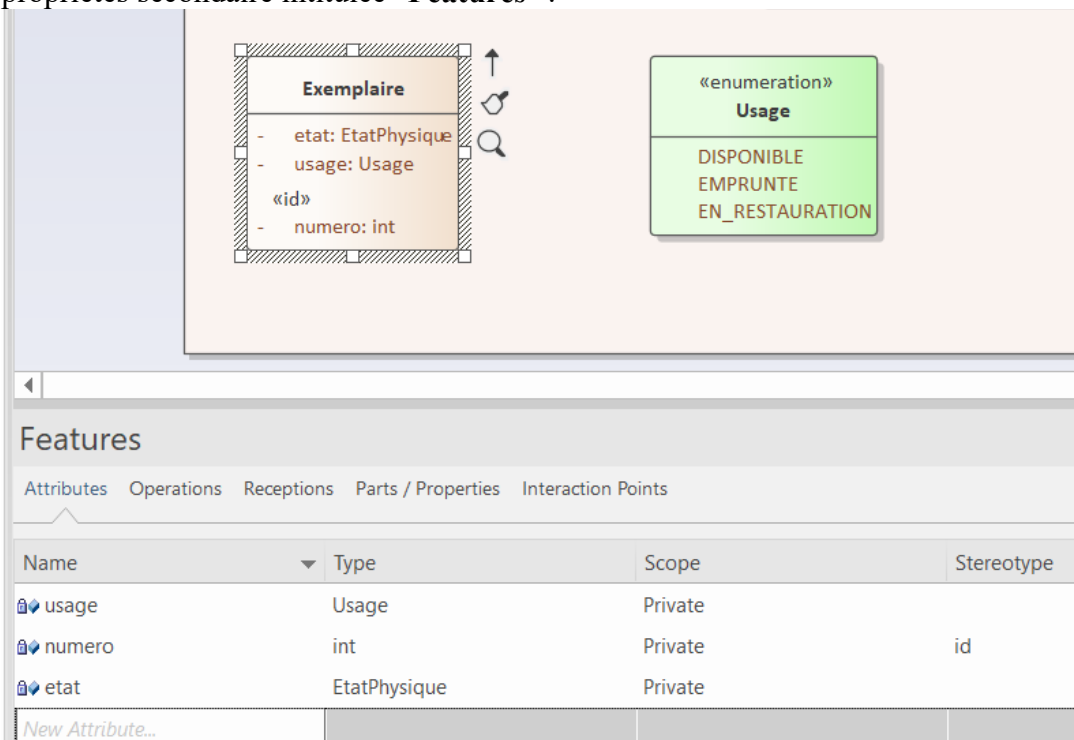
- Effectuer ensuite un glisser/poser du package (de l'arborescence vers le diagramme) .
- Souvent besoin d'ajouter des dépendances entre les packages
- Eventuel ajout de classes dans les packages
- En plus du diagramme de classes principal "main class diagram" , généralement un diagramme de classes (plus détaillé) par package .



Selon les préférences --> click droit sur fond d'un diagramme et **properties .../ diagram /** cocher ou décocher **"show namespace"** ou autres .
et **properties .../elements/show compartments/** cocher ou décocher **"package contents"** ou autres .

1.8. Edition d'un diagramme de classes

Beaucoup de paramétrages fondamentaux (ex : liste des attributs, ...) s'effectuent via la fenêtre de propriétés secondaire intitulée **"Features"** :



On peut faire apparaître cette fenêtre via un **click droit / features ... / attributes ...** ou autre .

En analyse fonctionnelle (avec conception détaillée) , on pourra quelquefois ne pas spécifier le type de données via le choix "**none**" .

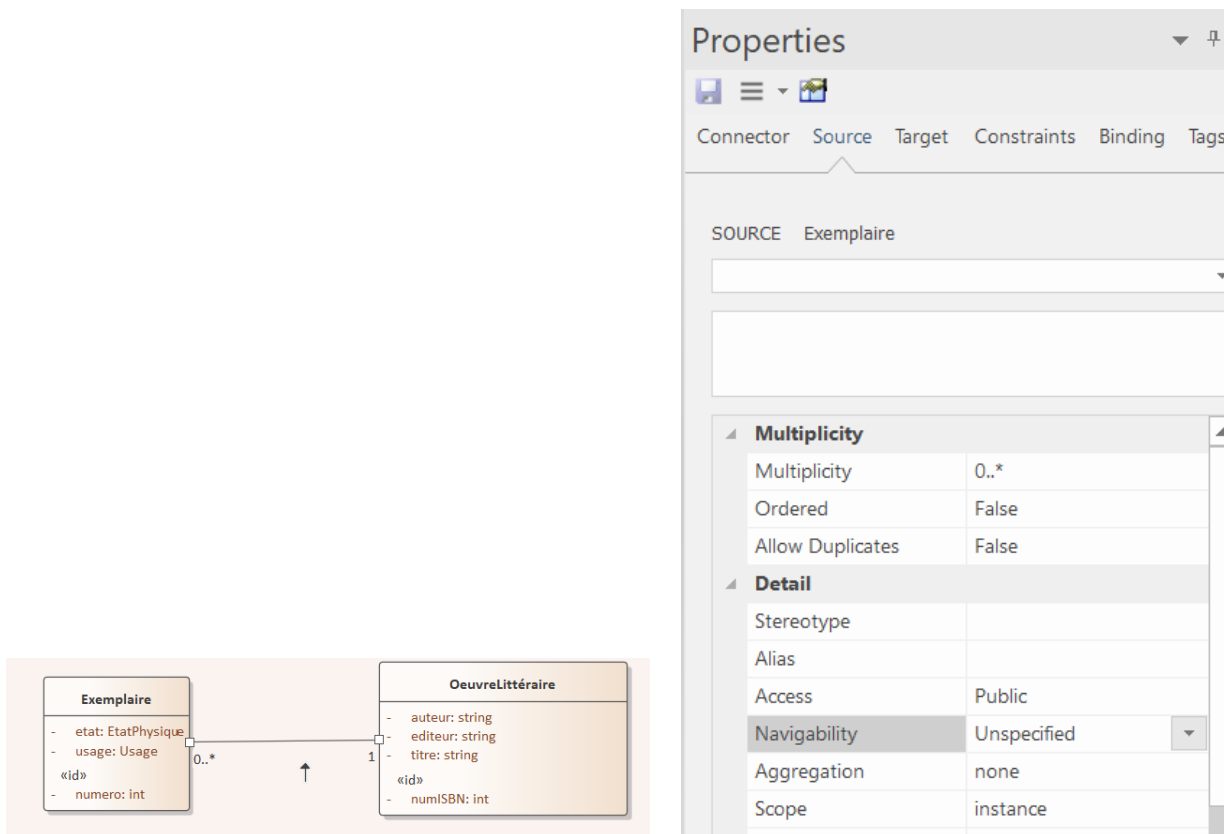
Certains nom de types (manquants) peuvent être créés à la volée (ex : "**string**" , "**number**" , ...) .

Via **Select Type ...** on peut choisir une de nos classes (ou énumération) comme type de données d'un attribut à paramétrer .

Les attributs sont (par défaut) affichés par ordre alphabétique .

Un stéréotype de type <<id>> (ou <<pk>>) peut éventuellement servir à préciser l'identifiant (ou la clef primaire) .

De façon à paramétrer les extrémités des associations (rôles , multiplicités , navigabilité, ...) , on pourra (entre autres) utiliser les onglets "**Source**" et "**Target**" de la fenêtre de propriétés :



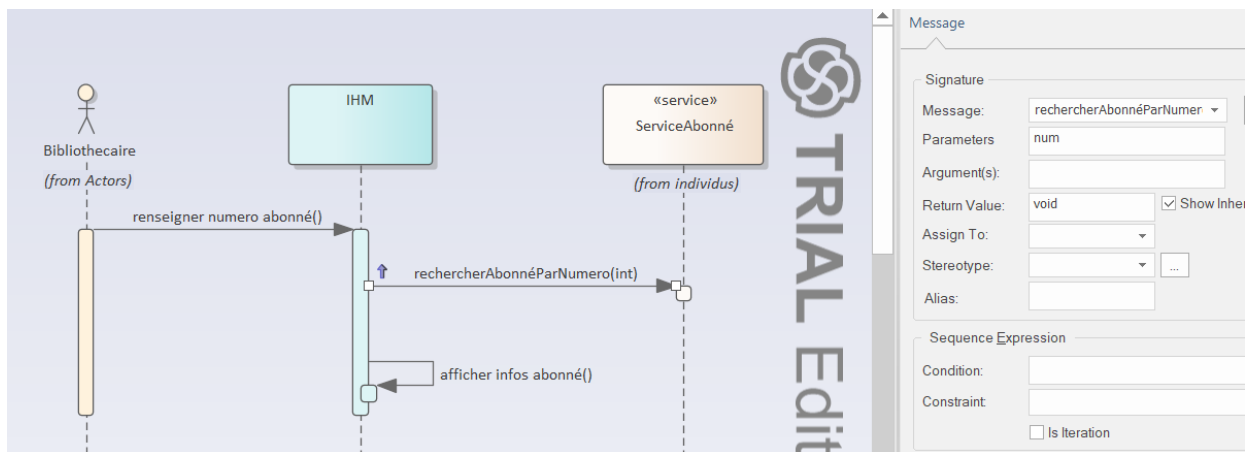
1.9. Edition d'un diagramme d'états

Tout est assez intuitif .

Dans la partie "Constraints" , la partie "effect : " permet de saisir une action (courte) déclenchée lors d'un changement d'état (transitions).

1.10. Edition d'un diagramme de séquence

- Sélectionner un "use case" dans l'arborescence.
- Ajouter un nouveau diagramme de séquence (via add / diagram.../ behavior / sequence diag).
- placer les acteurs et les classes existantes concernées via des "glisser/poser" depuis l'arborescence du modèle .
- placer les autres "lifeLine" via la "toolbox"
- retranscrire ensuite un scénario (1,2,3...) en une suite/cascade d'envois de messages .

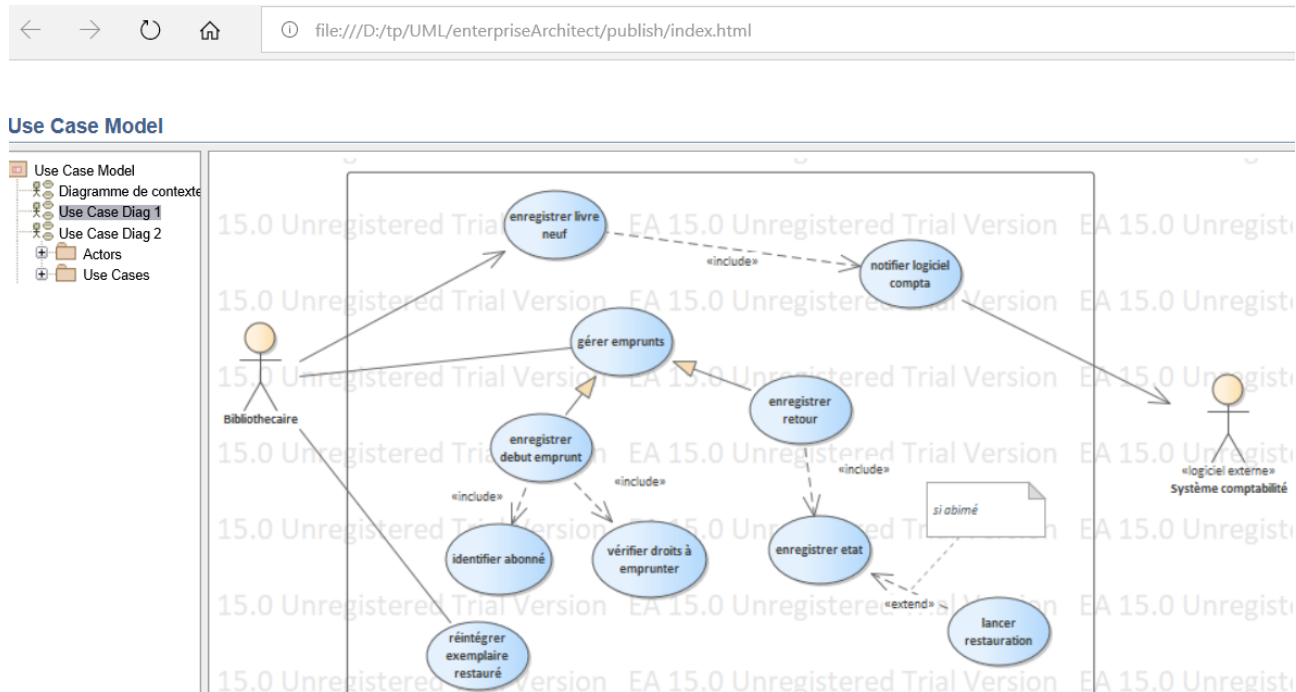


1.11. Edition d'un diagramme de déploiement

Les "cubes" c'est un jeu d'enfants , c'est intuitif !

1.12. Exportation d'un modèle UML entier sous forme de site HTML facilement consultable à distance

le menu **Publish / HTML / Web report / standard Html report** permet de générer automatiquement un paquet de pages HTML (à publier sur un serveur HTTP) correspondant à une bonne description arborescente du modèle UML avec diagrammes incorporés et navigations hypertextes .



NB : le paquet de pages html générées utilise en interne un peu de javascript pour naviguer d'un élément à un autre.

Selon les réglages (par défaut) des navigateurs "internet" , une visualisation directe (en file:///) sera quelquefois refusée . On pourra alors essayer une visualisation avec un autre navigateur ou bien lancer un mini serveur http (exemple: *lite-server* et test via <http://localhost:3000/index.htm>[1])