

HOPEX Application Design

Guide d'utilisation

HOPEX V2R1



Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis et ne sauraient en aucune manière constituer un engagement de la société MEGA International.

Aucune partie de la présente publication ne peut être reproduite, enregistrée, traduite ou transmise, sous quelque forme et par quelque moyen que ce soit, sans un accord préalable écrit de MEGA International.

© MEGA International, Paris, 1996 - 2018

Tous droits réservés.

Mega IT Designer et HOPEX sont des marques réservées de MEGA International.

Windows est une marque réservée de Microsoft.

Les autres marques citées appartiennent à leurs propriétaires respectifs.

INTRODUCTION



HOPEX Application Design s'appuie sur les outils proposés par la plateforme **HOPEX** pour assister les architectes dans la spécification des évolutions de leur système d'information. Les fonctionnalités avancées du produit, décrites ici, sont illustrées par l'exemple d'une application de traitement des demandes d'achat.

Voici les points abordés dans **HOPEX Application Design** :

- ✓ "Créer un projet de conception d'application", page 31 ;
- ✓ "Définir un projet AD et son périmètre applicatif", page 31 ;
- ✓ "Définir les objectifs et exigences d'un projet (AD)", page 43 ;
- ✓ "Décrire les spécifications fonctionnelles d'un projet", page 53 ;
- ✓ "Décrire les données d'un projet", page 67 ;
- ✓ "Décrire les spécifications techniques d'un projet", page 77.

Pour plus de détails sur l'interface et les fonctionnalités **HOPEX** en général, voir :

- ✓ "Présentation de HOPEX Application Design", page 2 ;
- ✓ "La méthode d' HOPEX Application Design", page 5 ;
- ✓ "Connexion au bureau d'HOPEX Application Design", page 21.

Pour commencer avec **HOPEX Application Design**, voir :

- ✓ "Créer un projet de conception d'application", page 32 ;

PRÉSENTATION DE HOPEX APPLICATION DESIGN

Associé à l'ensemble des produits de la suite **HOPEX**, **HOPEX Application Design** offre une méthodologie et les outils qui permettent de planifier votre transformation métier.

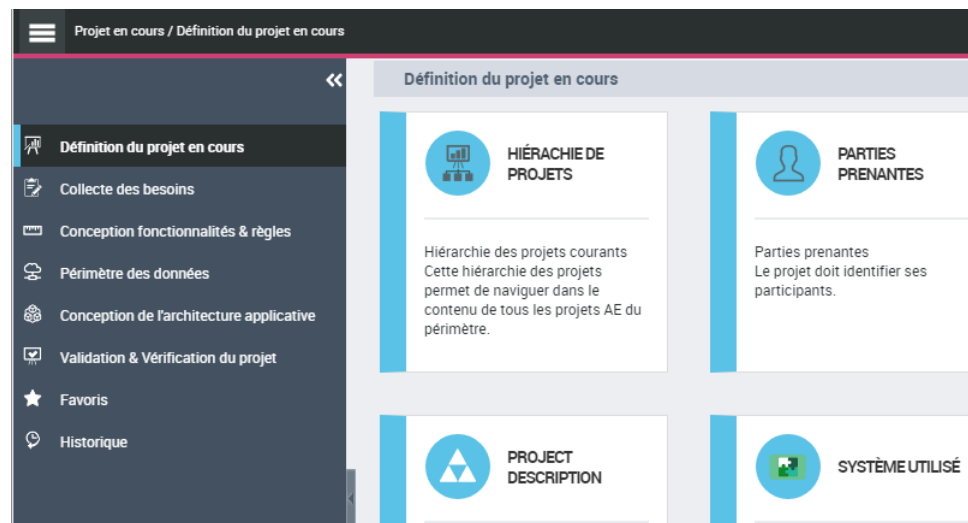
Le périmètre couvert par HOPEX Application Design

Pour assister les architectes dans la description des spécifications logicielles, la solution **HOPEX Application Design** s'appuie sur la méthodologie suivante :

- Définition du projet et de son périmètre applicatif
- Capture des besoins métier et objectifs du projet
- Analyse fonctionnelle et description des contextes d'utilisation
- Modélisation des flux de données applicatifs (périmètre des données)
- Conception de l'architecture technique

Un workflow de validation offre au client final la possibilité de valider les spécifications en s'assurant que ses besoins sont satisfaits.

L'interface de la solution suit cette méthodologie. Pour chaque étape, la solution fournit aux utilisateurs des outils spécifiques.



Les phases proposées en standard s'intègrent dans une approche de développement Agile.

Des rapports d'analyse sont proposés pour faciliter l'analyse du sujet et contribuer à l'élaboration de spécifications de développement complètes et pertinentes.

Les développements, l'intégration et les tests sont réalisés en dehors des phases du cycle de projet géré avec **HOPEX Application Design**.

Les profils de HOPEX Application Design

Dans **HOPEX Application Design**, il existe, par défaut, plusieurs profils auxquels sont associés des droits et accès. Ces profils sont :

- l'**Administrateur fonctionnel de conception d'application**,
- le **Concepteur d'application**,
- le **Propriétaire d'application**,
- le **Concepteur UML**,
- le **Contributeur de conception d'application**.

L'administrateur fonctionnel de conception d'application

L'administrateur fonctionnel de conception d'application est responsable de la gestion de l'ensemble de l'aspect fonctionnel de la conception d'application. Il possède des droits étendus sur tous les objets gérés. Il a aussi en charge l'organisation du travail des concepteurs d'application.

- Il gère la création des utilisateurs et leur assignation aux profils ;
- Il crée les projets et identifie les objets du référentiel **HOPEX** qui font partie du périmètre ;
- Il crée les plans d'entreprise et identifie les objets du référentiel **HOPEX** qui font partie du périmètre ;
- Il spécifie les intervenants dans le plan d'entreprise ainsi que le rôle de chacun ;
- Il affecte les utilisateurs à l'un ou l'autre des projets qui constituent les points d'entrée dans le bureau **HOPEX Application Design**.

➡ Pour plus de détails sur le bureau de l'administrateur fonctionnel, voir ["Présentation du bureau de l'administrateur fonctionnel de conception d'application"](#), page 24.

Le concepteur d'application

Le concepteur d'application est le profil utilisateur de la solution **HOPEX Application Design** chargé de conduire des projets de conception d'application.

Il définit les environnements des applications, leurs structures, les scénarios d'interaction, les processus applicatifs et les interfaces.

Si votre licence le permet, et afin que les utilisateurs reliés à ce profil puissent intégrer leurs travaux, le concepteur d'application peut aussi accéder aux objets et aux fonctionnalités principales des solutions **HOPEX Business Process Analysis**, **HOPEX IT Architecture**, **HOPEX IT Portfolio Management** et **HOPEX Risk Mapper** au travers du bureau **HOPEX Business Architecture**.

➡ Pour plus de détails sur le bureau du concepteur d'application, voir ["Présentation du bureau du concepteur d'application"](#), page 22.

Le propriétaire d'application

Le propriétaire d'application est chargé de spécifier les caractéristiques des applications dont il est responsable et de les mettre à jour.

☛ Pour plus de détails sur le bureau du concepteur d'application, voir ["Présentation du bureau du Propriétaire d'application", page 24.](#)

Le concepteur UML

Le concepteur UML est un profil utilisateur technique de la solution **HOPEX Application Design**.

Il est chargé des spécifications détaillée du système et de la construction des diagrammes UML.

☛ Pour plus de détails sur le bureau du concepteur d'application, voir ["Présentation du bureau du Concepteur UML", page 25.](#)

Le contributeur de conception d'application

Le contributeur de conception d'application est le profil utilisateur final métier de la solution **HOPEX Application Design**. Il n'accède à l'application qu'en lecture pour construire des rapports.

☛ Pour plus de détails sur le bureau du concepteur d'application, voir ["Présentation du bureau du Contributeur de conception d'application", page 25.](#)

Les rôles métier de HOPEX Application Design

Dans **HOPEX Application Design**, il existe, par défaut, des *rôles métier* qui peuvent être assignés à certains utilisateurs.

📖 Un rôle métier permet d'assigner une tâche à une personne (ex. : décrire une application) et si besoin pour une localisation spécifique (ex. : site de paris). Des rôles métier sont assignés à des personnes et une personne peut avoir plusieurs rôles métier.


Ces rôles métier sont :

- **Concepteur de données** qui est responsable de la gestion des données.
- **Propriétaire local d'application**, son rôle est de renseigner les caractéristiques des applications dont il a la responsabilité, et de les mettre à jour régulièrement.
- **Concepteur de logiciel**, son rôle est de renseigner les caractéristiques fonctionnelles des logiciels dont il a la responsabilité, et de les mettre à jour régulièrement.

☛ Pour plus de détails sur l'utilisation des rôles métier dans **HOPEX Application Design**, voir ["Relier une personne à une organisation", page 37.](#)

LA MÉTHODE D' HOPEX APPLICATION DESIGN

La méthode utilisée dans **HOPEX Application Design** s'appuie sur la notion de projet.

 *Un projet d'Architecture d'Entreprise est une partie d'un système dont l'étude est confiée à une même équipe, qui transforme un système, ou une partie d'un système, en vue d'atteindre un objectif donné.*

Présentation de la méthode et exemple

La spécification des projets d'évolution du SI avec **HOPEX Application Design** comprend les étapes suivantes :

- 】 **Décrire le périmètre du projet** : cette première étape consiste à définir le périmètre applicatif du projet, ses livrables, la décomposition éventuelle en sous-projets et l'organisation des intervenants sur le projet.
- 】 **Décrire les objectifs et les exigences du projet**
- 】 **Construire les spécifications fonctionnelles du projet** : cette étape consiste à définir les fonctionnalités attendues et le contexte d'utilisation des applications du projet.
- 】 **Décrire les données du projet** : il s'agit de décrire la façon dont les données d'une organisation sont utilisées par les processus et les applications.
- 】 **Construire les spécifications techniques du projet** : les spécifications techniques du projet visent à détailler l'architecture technique des composants du projet.
- 】 **Décrire les échanges de données** : il s'agit de décrire les contrats d'échange entre les composants d'une l'architecture métier ou informatique.

Présentation de l'exemple

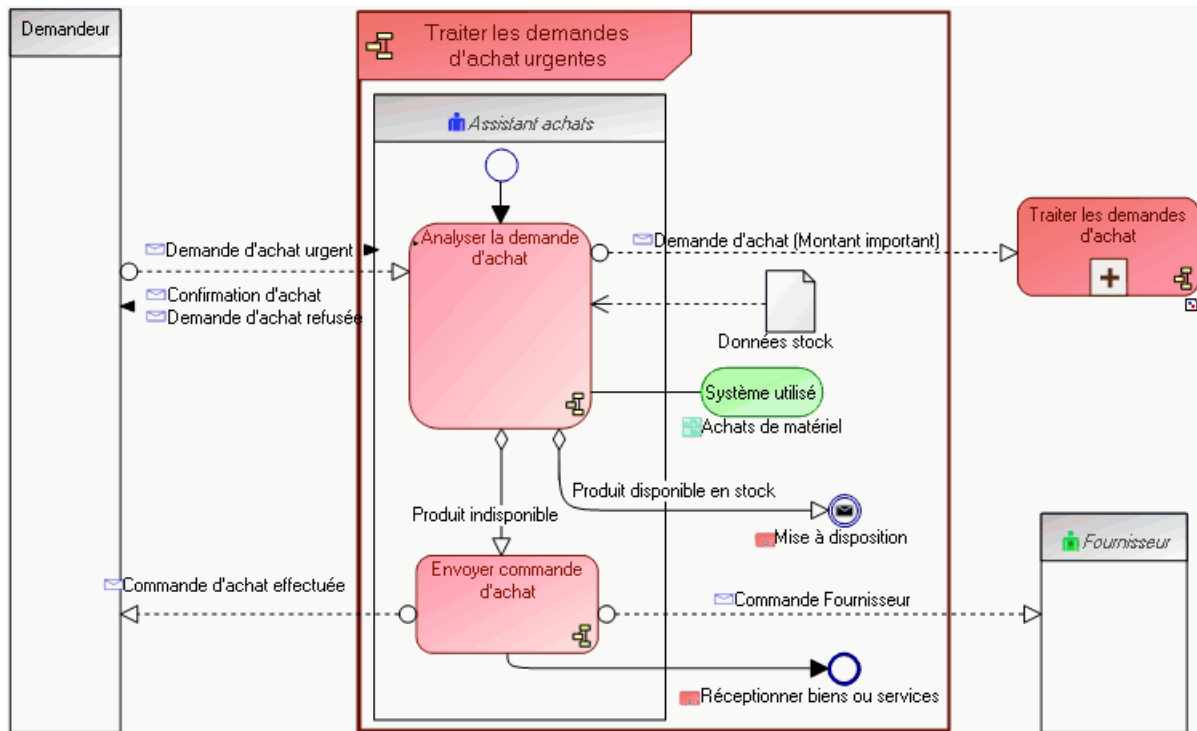
Ces différents points sont illustrés par l'exemple d'une entreprise de location de bateaux qui a pris la décision d'améliorer son processus d'achats.

La disponibilité des bateaux étant un facteur essentiel à l'activité de l'entreprise, un processus spécifique doit être défini pour traiter les achats urgents du service d'entretien des véhicules.

Les processus des achats de l'entreprise se répartissent entre les processus d'achats opérationnels et les processus d'achat d'investissement. Les processus d'achats opérationnels incluent les achats de pièces détachées et de matière première destinés à l'entretien et à la réparation des bateaux.

Présentation du contexte organisationnel

Le nouveau processus organisationnel prévu pour le traitement des demandes d'achat urgentes vise à étendre les responsabilités de l'assistant achats afin d'envoyer la commande d'achat dans des délais très courts.



Processus organisationnel "Traiter les demandes d'achat urgentes".

L'assistant achat commence par analyser la demande d'achat. Si le montant est important, le processus normal de traitement des demandes d'achat est mis en oeuvre.

Si le produit est disponible, l'assistant envoie la demande de mise à disposition.

Sinon, l'assistant envoie la commande d'achat au fournisseur. Le reste du traitement est effectué dans le cadre du processus "Traiter les demandes d'achat" normal.

Pour lui permettre d'analyser la demande d'achat et de passer la commande rapidement, il est prévu de mettre à la disposition de l'assistant achat une nouvelle application lui donnant accès aux données concernant les stocks et les fournisseurs.

Décrire un projet et son environnement

Le *projet* est l'élément de base de l'utilisation de **HOPEX Application Design**.

Un projet d'Architecture d'Entreprise est une partie d'un système dont l'étude est confiée à une même équipe, qui transforme un système, ou une partie d'un système, en vue d'atteindre un objectif donné.

☛ Pour plus de détails sur la création d'un projet, voir "[Créer un projet de conception d'application](#)", page 32.

Décrire le périmètre du *projet* implique de réaliser les tâches suivantes :

- décrire les applications du projet et leurs échanges
- décrire les livrables du projet

Avec **HOPEX Application Design**, vous pouvez également définir les différentes organisations concernées par le projet.

☛ Pour plus de détails sur la gestion des organisations concernées par le projet, voir "[Décrire l'organisation des participants au projet](#)", page 36.

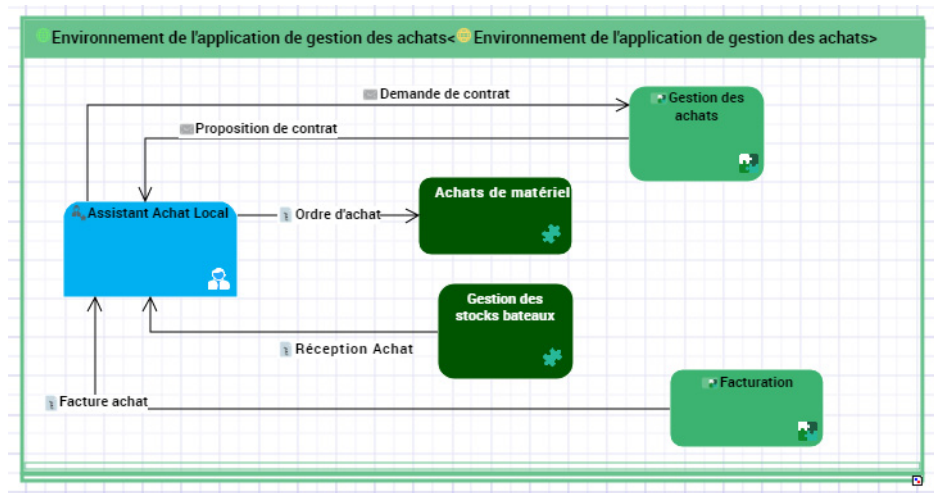
Décrire les applications du projet et leurs échanges

Un environnement d'application permet de représenter un contexte d'utilisation d'une application.

Avec **HOPEX Application Design**, les diagrammes de scénario de flux permettent de décrire les flux échangés par une application, un service applicatif ou un micro service dans un contexte particulier.

☛ Pour plus de détails sur les diagrammes de scénario de flux, voir "[Décrire les scénarios de flux](#)", page 154.

Le diagramme ci-dessous présente les flux échangés dans le contexte de l'environnement de l'application "Système de Gestion des achats".



Les achats de matériel destinés à l'entretien des bateaux sont effectués directement par le bureau de représentation en utilisant un service applicatif "Achats de matériel". Celui-ci fait appel à une application centralisée de "Gestion des achats" au siège.

S'il s'agit d'un nouveau type de matériel ou d'un nouveau fournisseur, une demande de contrat est envoyée au fournisseur à l'aide de l'application du siège. Celui-ci renvoie en échange une proposition de contrat qui est soumise au demandeur.

Dans les autres cas, un agent du bureau de représentation envoie un ordre d'achat au fournisseur à l'aide de l'application "Achats de matériel".

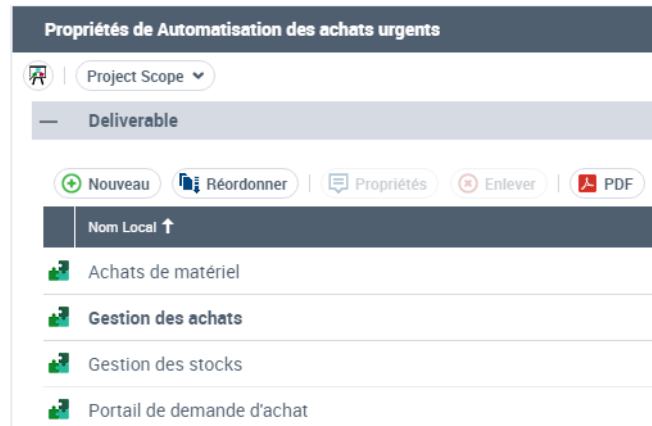
La réception des achats est réalisée à l'aide de l'application "Gestion des stocks bateaux" du bureau de représentation.

La facture est traitée par l'application "Facturation" du siège.

Décrire les livrables du projet

Après analyse, les éléments applicatifs qui seront affectés par le projet sont décrits dans les livrables du projet.

✎ Pour plus de détails sur la mise à jour des livrables d'un projet, voir ["Décrire les scénarios de flux", page 154](#).



L'application "Achat de matériel" est la principale application impactée par le projet, en particulier pour l'amélioration de son portail d'accès.

L'application "Gestion des stocks" est utilisée par l'application "Achat de matériel" pour consulter et mettre à jour les informations concernant l'état des stocks et, en particulier, celui des pièces détachées.

L'application "Gestion des achats" utilise la base de données "Catalogue" pour passer une commande d'achat.

L'application "Gestion Entretien et Réparation" et l'application "Gestion des pièces de rechange" vont également utiliser le portail de l'application "Achats de matériel".

Pour plus de détails sur la description d'un environnement de projet, voir ["Définir un projet AD et son périmètre applicatif", page 31](#).

Décrire les objectifs et les exigences d'un projet

Afin d'affiner la description du périmètre technique d'un *projet*, cette étape consiste à collecter les *objectifs* et les *exigences* des différentes catégories d'utilisateurs.

📖 Un objectif est un but que l'on cherche à atteindre ou la cible visée par un processus ou une opération. Il permet de mettre en évidence les points que l'on veut améliorer pour ce processus ou cette opération.

📖 Une exigence est un besoin ou une attente formulés explicitement, imposés comme une contrainte à respecter dans le cadre d'un projet de certification, d'organisation ou de modification du système d'information d'une entreprise.

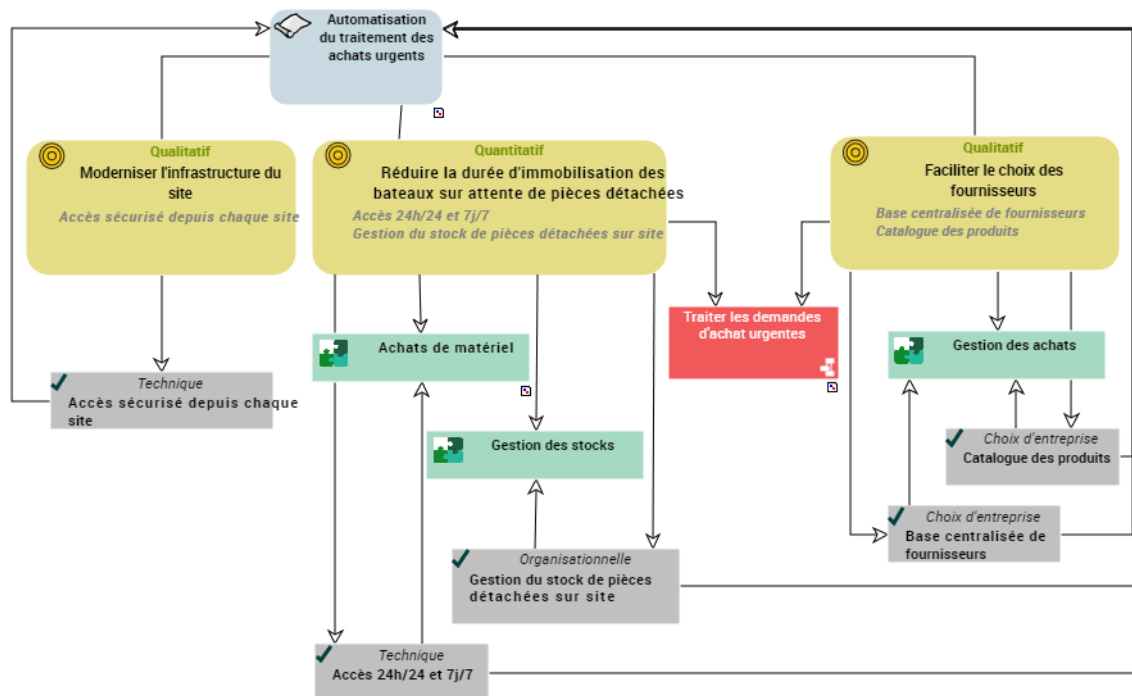
Dans notre exemple, le développement d'une application supportant le traitement des demandes d'achat urgentes vise plusieurs objectifs :

D'un point de vue fonctionnel, cette application doit permettre de réduire la durée d'immobilisation des bateaux due à un défaut de pièces détachées. Le traitement des demandes d'achat doit donc être accessible 24h sur 24 parce que les bureaux de représentation de l'entreprise sont répartis dans différents continents et 7 jours sur 7 puisque l'entreprise intervient dans le métier du tourisme.

Pour limiter les délais liés à la recherche de fournisseurs de pièces détachées déjà référencées ou non, le département achat a décidé de mettre en place un catalogue de produits et une base centralisée de fournisseurs.

Enfin, ce projet répond à un objectif technique de modernisation et de sécurisation des accès entre le siège de l'entreprise et les bureaux de représentation.

Le diagramme d'objectifs et d'exigences d'un projet permet de représenter et de classer les éléments recensés.



Pour plus de détails sur la gestion des objectifs et exigences, voir "[Définir les objectifs et exigences d'un projet \(AD\)](#)", page 43.

Construire les spécifications fonctionnelles

Après avoir défini le périmètre, les objectifs et exigences du projet, cette étape consiste à décrire les fonctionnalités attendues du futur système par les différentes catégories d'utilisateurs.

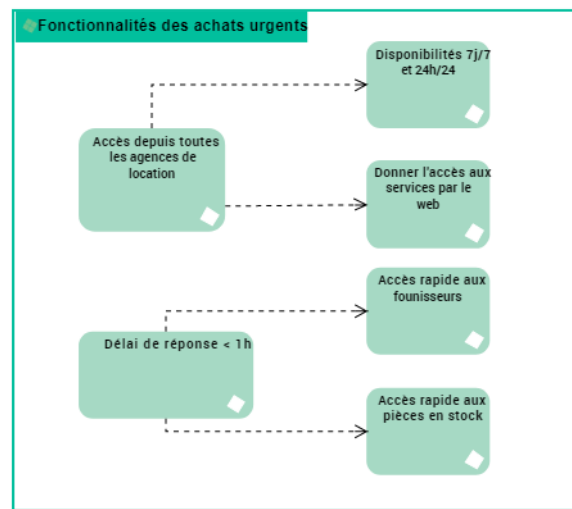
Décrire la carte des fonctionnalités attendues du projet

Une carte de fonctionnalités est un assemblage de fonctionnalités avec leurs dépendances qui, conjointement, définissent le périmètre d'une architecture matérielle ou logicielle.

Décrire la *carte de fonctionnalités* permet de dresser la liste des fonctionnalités attendues du projet et d'associer chacune d'elles à un objectif ou une exigence, d'une part, et à un élément applicatif d'autre part.

➡ Pour plus de détails sur la gestion des fonctionnalités attendues du projet, voir "[Décrire les fonctionnalités d'un projet](#)", page 54.

Dans notre exemple, une carte de fonctionnalités se présente sous la forme du diagramme suivant.



Les fonctionnalités attendues du projet de traitement des achats urgents sont relatives au temps de réponse et à la disponibilité de l'application depuis les agences de locations. Les fonctionnalités principales sont déclinées en sous-fonctionnalités.

Décrire les environnements d'application du projet

Un environnement d'application présente le contexte d'utilisation des applications d'un projet. Il décrit les interactions, entre les acteurs impliqués et les applications internes et externes du projets, qui permettent d'assurer les fonctionnalités attendues du projet.

Plusieurs types de diagrammes sont proposés pour décrire un *environnement d'application* :

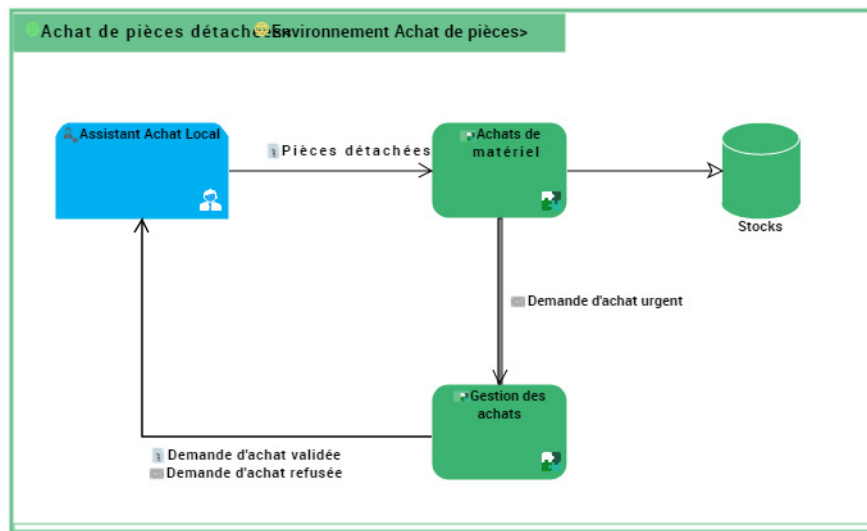
- Le diagramme de scénario de flux d'environnement d'application permet de décrire les flux échangés entre une application et son environnement entre dans un contexte donné.
- Le diagramme d'environnement d'application permet de décrire les interactions entre une application et son environnement décrit : ses utilisateurs, les applications et les services externes.
- Le diagramme de cas d'utilisation d'environnement d'application permet de décrire au format UML les cas d'utilisation de l'environnement d'application.

☛ Une même application peut être associée à plusieurs *environnement d'application* afin de représenter plusieurs contextes d'utilisation de l'application.

Exemple de diagramme de scénario de flux

Un diagramme de scénario de flux peut être construit pour un environnement d'application, une application, un service applicatif ou un micro-service.

☛ Pour plus de détails sur la construction d'un diagramme de scénario de flux, voir "[Décrire les scénarios de flux](#)", page 154.



Ce diagramme de scénario de flux décrit l'environnement de l'application "Achat de pièces détachées".


La demande de "pièce détachée" est envoyée par l'assistant Achat local à l'application "Achat de matériel".

L'application "Achat de matériel" consulte les stocks. Si la pièce n'est pas en stock, une "demande d'achat urgent" est envoyée à l'application "Gestion des achats".

L'application "Gestion des achats" valide ou refuse la demande.

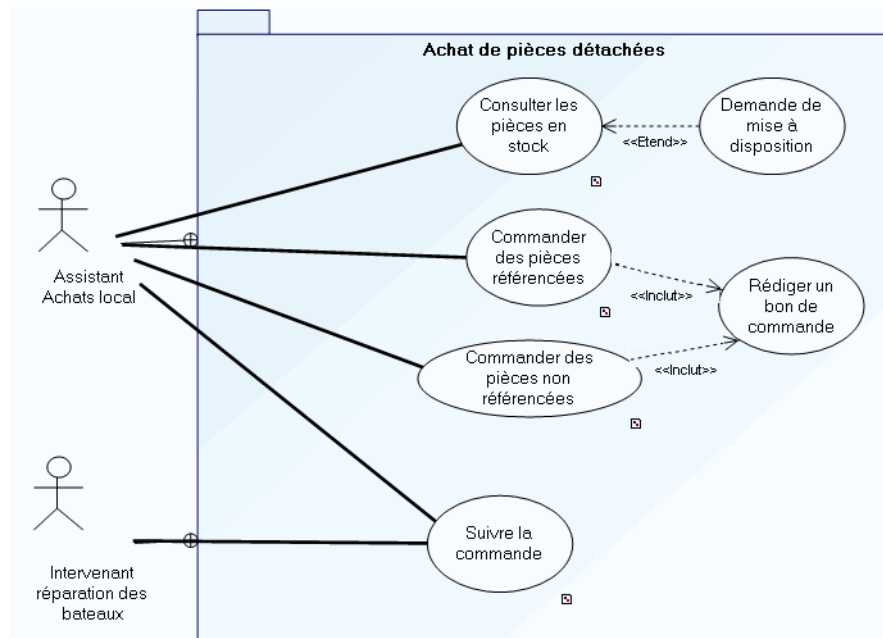
Exemple de diagramme de cas d'utilisation

Vous pouvez également utiliser un diagramme de **cas d'utilisation** pour décrire les interactions entre un élément applicatif et les acteurs de l'organisation dans le contexte précis.

 Un cas d'utilisation est une suite d'actions qui amène un résultat observable pour un acteur particulier. Des scénarios illustrent les cas d'utilisation par l'exemple.

Les diagrammes de cas d'utilisation peuvent être construits pour des environnements d'application, des applications, des services applicatifs et des micro-services.

➡ Pour plus de détails sur la création d'un cas d'utilisation, voir ["Décrire un cas d'utilisation d'un projet", page 62.](#)



Le système est utilisé pour la consultation des pièces en stock et la commande de nouvelles pièces détachées.

La consultation des pièces en stock est effectuée par l'assistant achat local. Suite à la consultation, l'assistant peut faire une demande de mise à disposition.

Deux types de commandes sont possibles; une commande de pièces déjà référencées ou une commande de pièces non référencées. Dans les deux cas, un bon de commande doit être rédigé.

Le suivi de la commande est assuré à la fois par l'assistant achat local et l'intervenant chargé de la réparation des bateaux.

Exemple de diagramme d'environnement d'application

➡ Pour plus de détails sur la construction d'un diagramme d'environnement d'application, voir "[Décrire un environnement d'application](#)", page 58.

Le diagramme suivant décrit l'environnement d'application correspondant au traitement des achats de pièces détachées.

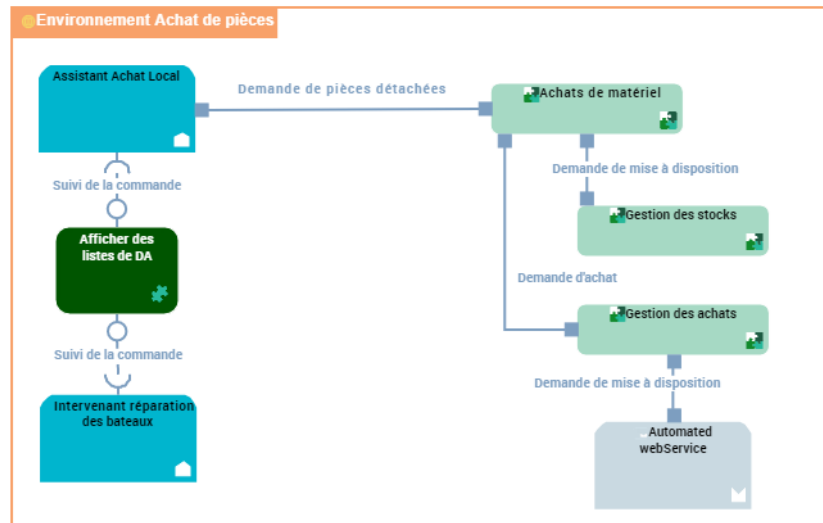


Diagramme d'environnement d'application "Achat de pièces détachées"

Les demandes d'achat de pièces détachées sont formulées par les intervenants sur les réparations de bateaux et elles sont traitées par les assistant achat locaux.

La consultation des pièces en stock est effectuée par l'assistant achat local. Suite à la consultation, l'assistant peut faire une demande de mise à disposition.

Deux types de commandes sont possibles; une commande de pièces déjà référencées ou une commande de pièces non référencées. Dans les deux cas, une demande de mise à disposition est effectuée.

Le suivi de la commande est assuré à la fois par l'assistant achat local et l'intervenant chargé de la réparation des bateaux.

Pour plus de détails sur les spécifications fonctionnelles, voir "[Décrire les spécifications fonctionnelles d'un projet](#)", page 53.

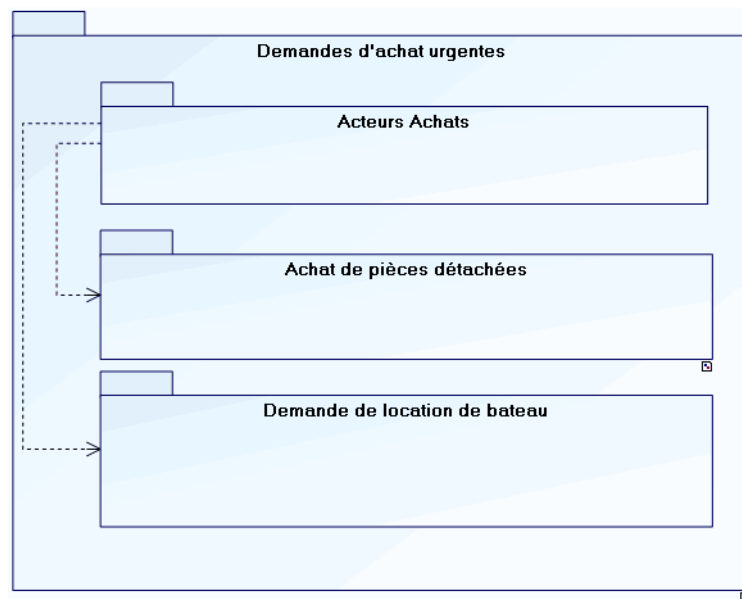
Décrire les données du projet

Cette étape a pour objectif de décrire les *domaines de données* afin de déduire la structure des bases de données qui seront mise en oeuvre par le projet.

Pour mener à bien cette étape, nous vous proposons de réaliser les tâches suivantes.

Spécifier les paquetages

Le paquetage vous permet de classer les éléments référencés dans un projet. Vous pouvez créer des sous-paquetages dans un paquetage afin de classer plus finement les objets, par exemple les acteurs d'un projet.



Les demandes d'achat urgentes sont prévues pour traiter les achats de pièces détachées et les demandes de location de bateau. Dans ces deux cas, les utilisateurs sont des acteurs du domaine des achats.

Spécifier les domaines de données logiques

Un domaine de données logique permet de définir une structure de données logiques constituée de classes et de vue de classes.

Un domaine de données logique est détenu par un paquetage, il peut référencer des objets détenus dans d'autres paquetages.

Lors d'une intégration, une structure physique correspondante peut être définie via un domaine de données physique. Celui-ci est constitué de tables et de vues de tables.

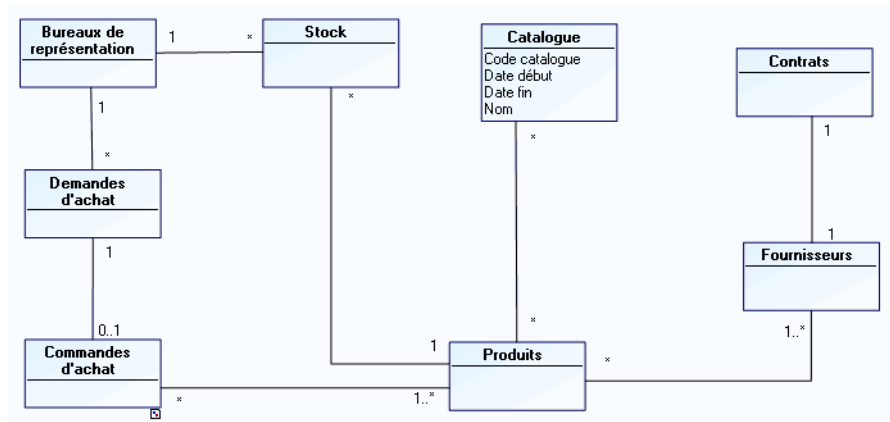
Le diagramme de domaine de données logiques suivant représente une structure de données relative aux Commandes.

Spécifier les domaines de données relationnelles

Un domaine de données relationnelles permet de positionner les bases de données dans le modèle logique de gestion des données.

Le modèle de données relationnelles du projet "Automatisation des demandes d'achat" est présenté ci-dessous.

L'application gère les demandes d'achat, les commandes et



les stocks de produits dans chacun des bureaux de représentation.

Un catalogue centralisé des produits et des fournisseurs est mis en place.

Les contrats avec les fournisseurs référencés sont également accessibles depuis l'application.

Pour plus de détails sur la description des données, voir ["Décrire les données d'un projet", page 67](#).


Construire les spécifications système


Après les spécifications fonctionnelles et la description des données utilisées, les spécifications techniques du projet visent à détailler l'architecture cible du projet.


Décrire les éléments de l'architecture cible

Après les spécifications fonctionnelles et la description des données utilisées, les spécifications techniques du projet visent à décrire l'architecture technique des

composants du projet : *applications, services applicatifs, micro-services, bases de données.*

 Une application est un composant logiciel déployable qui fournit un ensemble de fonctionnalités à des utilisateurs.

 Un service applicatif est l'élément de découpage d'une application qui est mis à la disposition de l'utilisateur final de cette application dans le cadre de son travail.

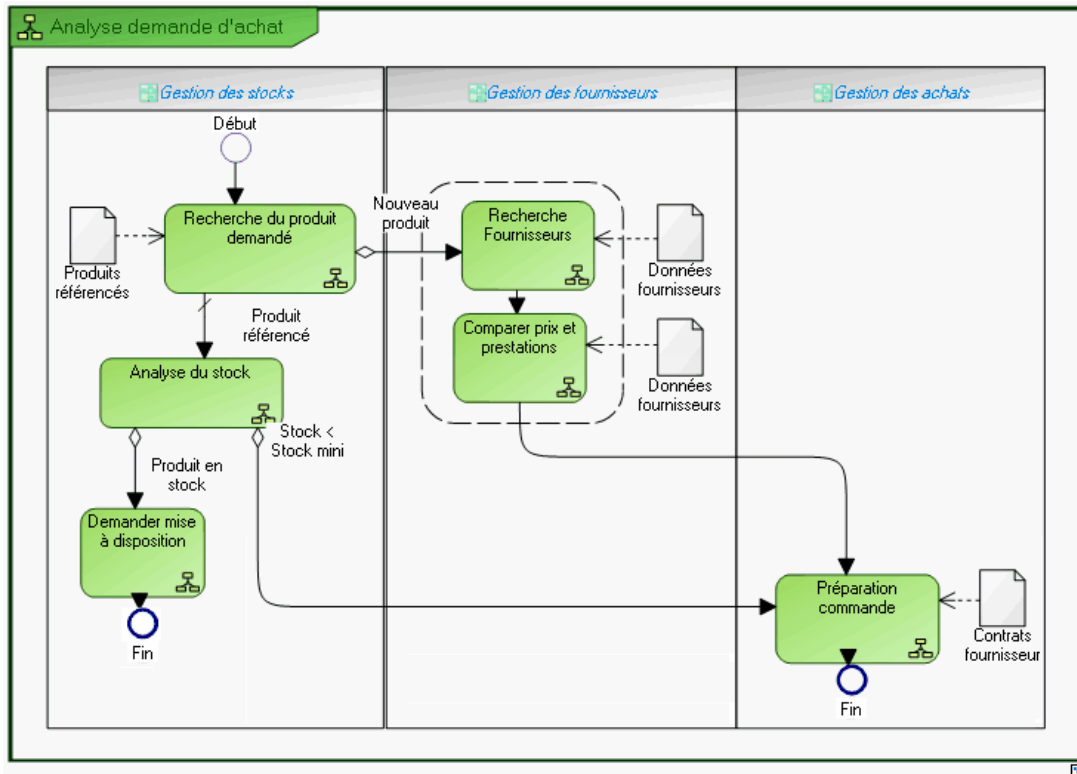
 Un micro-service est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

 Pour plus de détails sur la description de l'architecture cible, voir ["Décrire les services applicatifs et les micro-services", page 81.](#)

Présentation du processus applicatif

Un processus applicatif permet de décrire la séquence détaillée des tâches réalisées lors de l'exécution des services applicatifs.

Le diagramme de processus applicatif ci-après propose une première ébauche du fonctionnement attendu de la nouvelle application.



La recherche du produit est effectuée à partir de la base des produits référencés.

Si le produit est référencé, l'étude de l'état des stocks est réalisée.

Si le stock est suffisant, une demande de mise à disposition est activée et le processus prend fin.

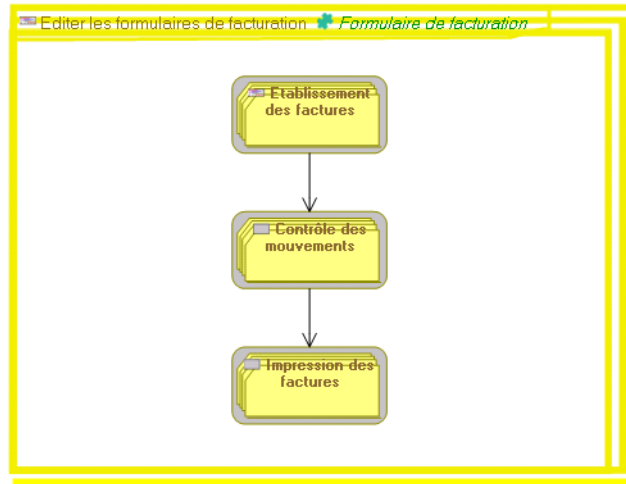
Si le stock est inférieur au stock minimum, une commande est émise vers le fournisseur.

Si le produit est nouveau, une recherche de fournisseur et une étude comparative des prix est menée. Puis une commande est émise et le processus prend fin.

➡ Pour plus de détails sur la description des processus applicatifs, voir ["Décrire les processus applicatifs", page 84.](#)

Décrire les traitements par lot

L'enchaînement de traitements automatisés peut être décrit dans un **diagramme de structure de chaîne de traitement de lot**.

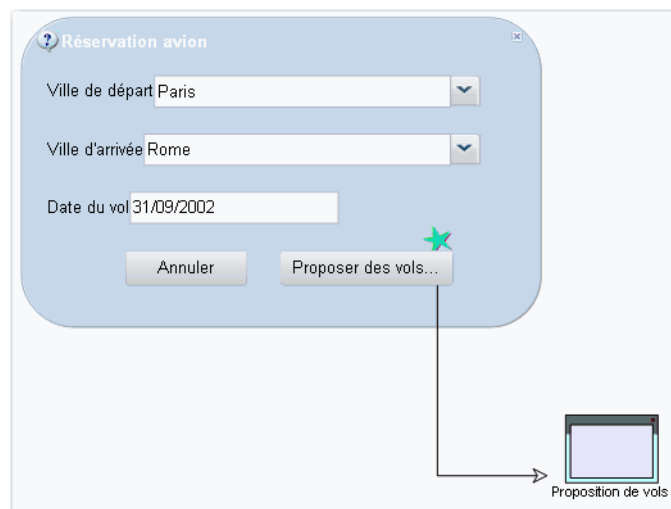


☛ Pour plus de détails sur la description des traitements par lot, voir ["Décrire les traitements de lot", page 91.](#)

Décrire la liste des services et des IHM

Il est possible de décrire les interfaces reliant les services ou les opérations avec l'extérieur. Cette description s'effectue dans un diagramme d'IHM.

Une diagramme d'IHM permet de décrire les IHM prévues.



☛ Pour plus de détails sur la description des IHM, voir ["Décrire les interfaces utilisateur", page 96.](#)

Pour plus de détails sur les spécifications techniques, voir "[Décrire les spécifications techniques d'un projet](#)", page 77.

CONNEXION AU BUREAU D'HOPEX APPLICATION DESIGN

☛ **HOPEX Application Design** s'adresse principalement à des utilisateurs web. Les bureaux décrits dans ce guide sont accessibles exclusivement aux utilisateurs de bureaux web.

Conditions préalables : importer les types élémentaires

Pour disposer des types élémentaires dans **HOPEX Application Design**, l'administrateur doit importer le solution pack "Information Architecture".

Préparer l'import

Les fichiers à importer sont livrés dans des fichiers compressés que vous devez décompresser avant d'importer dans un référentiel.

1. Dans le dossier où **HOPEX** est installé, ouvrez le dossier **Utilities**, puis le dossier **Solution Pack**.
2. Double-cliquez sur le fichier **Information Architecture.exe**.
3. Extrayez le contenu du fichier.

Importer le solution pack

Pour importer le solution pack :

1. Lancez "Administration.exe" et connectez-vous avec un utilisateur qui dispose de l'autorisation d'administration des données.

☛ L'identifiant "System" permet de se connecter avec l'utilisateur "Administrator". Cet utilisateur est créé par défaut avec les droits pour administrer les référentiels. Il n'a pas de Profil (il a tous les droits) et aucun mot de passe ne lui est attribué lors de l'installation.
2. Sélectionnez l'environnement puis le référentiel sur lequel vous souhaitez travailler.
3. Faites un clic droit sur le référentiel et sélectionnez **Gestion des objets > Importer un solution pack**.
La fenêtre de sélection du solution pack apparaît.
4. Sélectionnez **Information Architecture - DataType** et **Information Architecture - XsdType** et cliquez sur **OK**.
5. Quittez l'application d'administration.

Se connecter à la solution

Pour se connecter à **HOPEX Application Design**, voir **HOPEX Common Features**, "Le bureau HOPEX Web Front-End".

Le bureau d'HOPEX Application Design

Les menus et commandes disponibles dans **HOPEX Application Design** dépendent du profil avec lequel vous êtes connecté.

➡ Pour plus de détails sur l'utilisation de la plateforme Web des solutions HOPEX, voir le guide **HOPEX Common Features**.



Présentation de l'espace commun à tous les profils

L'ensemble des utilisateurs disposent du bureau de **HOPEX Application Design** et de l'accès aux volets suivants :

- **Accueil, Tableaux de bord et Liste de tâches** qui sont communs à tous les utilisateurs des solutions **HOPEX**.
- **Idéation** : désigne le processus de gestion des idées, qui deviendront d'éventuelles demandes de projet.
- **Rapports** : donne accès à l'ensemble des rapports qui permettent une meilleure compréhension des termes et de leur utilisation.

Présentation du bureau du concepteur d'application

Outre les volets proposés en standard à l'ensemble des utilisateurs du bureau de **HOPEX Application Design**, le concepteur d'application dispose des volets décrits ci-dessous.

Le volet Environnement

Le volet **Environnement** donne accès aux menus suivants :

- **Organisation**, pour accéder à la liste des acteurs et des processus du référentiels.
- **Navigation standard**, pour construire les Entreprises, affecter des utilisateurs aux différentes étapes.
- **Projets**, pour accéder à la liste des projets et en créer de nouveaux.

➡ Pour plus de détails sur la création de projets, voir "[Créer un projet de conception d'application](#)", page 31.

Le volet Projet en cours

Par défaut, lors de la connexion, ce volet n'est pas proposé.

Avant de définir un projet en cours, assurez-vous qu'un environnement de type "Conception d'application" a bien été défini dans les propriétés du projet en question. Voir "[Assigner un environnement de travail au projet](#)", page 32.

Pour définir le projet en cours :

1. Cliquez sur **Menu principal**.
2. Sélectionnez **Changer d'environnement de travail** et choisissez le projet préalablement créé sur lequel vous souhaitez travailler.

Le volet **Projet en cours** porte la méthode embarquée dans la solution **HOPEX Application Design** et il donne accès aux menus suivants :

- **Définition du projet en cours**, pour décrire le périmètre du projet, sa structure et les parties prenante.
☛ Pour plus de détails sur la création de projets, voir "Définir un projet AD et son périmètre applicatif", page 31.
- **Collecte des besoins**, pour décrire les objectifs et exigences du projet.
☛ Pour plus de détails sur la description des objectifs et exigences du projet, voir "Définir les objectifs et exigences d'un projet (AD)", page 43.
- **Conception fonctionnalités et règles**, pour construire les spécifications fonctionnelles du projet.
☛ Pour plus de détails sur les outils de spécification fonctionnelle, voir "Définir les objectifs et exigences d'un projet (AD)", page 43.
- **Périmètre des données**, pour décrire les paquetages, les données logiques et relationnelles du projet afin de définir les bases de données.
☛ Pour plus de détails sur les données du projet, voir "Décrire les données d'un projet", page 67.
- **Conception de l'architecture applicative**, pour décrire les éléments techniques du projet tels que les applications, les services applicatifs, les micro-services, les traitements par lot ou les IHM.
☛ Pour plus de détails sur les outils de spécification technique, voir "Décrire les spécifications techniques d'un projet", page 77.
- **Validation & Vérification**, cette facilité étape s'appuie sur des workflow qui permettent de gérer l'avancement du projet.

Voir aussi ["Créer un projet de conception d'application", page 31.](#)

Le volet Ressources des applications

Le volet **Ressources des applications** donne accès aux objets du référentiel référencés éventuellement dans le projet :

- **Processus**, voir le guide **HOPEX Business Process Analysis**.
- **Applications**, voir ["Décrire une application avec HOPEX Application Design", page 78.](#)
- **Contrats de service & Catalogues**, voir ["Décrire les échanges de données", page 163.](#)
- **Processus applicatifs, Lots et IHM**, voir ["Décrire les processus applicatifs", page 84.](#)
- **Technologies**, voir le guide **HOPEX IT Architecture**.
- **Piles de technologies**, voir le guide **HOPEX IT Architecture**.
- **Implémentation UML**, voir ["A propos de l'implémentation UML".](#)
- **Traitement de lots**, voir ["Décrire les traitements de lot", page 91.](#)
- **Architectures techniques**, voir ["Décrire les services applicatifs et les micro-services", page 81.](#)

Le volet Collaboration

Le volet **Collaboration** donne accès aux facilités de gestion des workflow.

Présentation du bureau de l'administrateur fonctionnel de conception d'application

L'administrateur fonctionnel de conception d'application de **HOPEX Application Design** dispose des volets de l'application proposés au concepteur d'application.

Il dispose en plus du volet **Administration** qui lui donne accès à la gestion des droits des utilisateurs.

➡ Pour plus de détails sur les tâches de l'administrateur, voir le guide d'administration **HOPEX Power Supervisor**.

Présentation du bureau du Propriétaire d'application

Outre les volets proposés en standard à l'ensemble des utilisateurs du bureau de **HOPEX Application Design**, le Propriétaire d'application dispose des volets suivants :

Le volet **Ressources des applications** qui donne accès aux objets du référentiel référencés éventuellement dans le projet :

- **Processus**, voir le guide **HOPEX Business Process Analysis**.
- **Applications**, voir ["Décrire une application avec HOPEX Application Design"](#), page 78.
- **Contrats de service & Catalogues**, voir ["Décrire les échanges de données"](#), page 163.
- **Processus applicatifs, Lots et IHM**, voir ["Décrire les processus applicatifs"](#), page 84.
- **Technologies**, voir le guide **HOPEX IT Architecture**.
- **Piles de technologies**, voir le guide **HOPEX IT Architecture**.
- **Implémentation UML**, voir ["A propos de l'implémentation UML"](#).
- **Traitement de lots**, voir ["Décrire les traitements de lot"](#), page 91.
- **Architectures techniques**, voir ["Décrire les services applicatifs et les micro-services"](#), page 81.

Le volet **Ressources des applications** donne accès aux objets du référentiel référencés éventuellement dans le projet :

- **Processus**, voir le guide **HOPEX Business Process Analysis**.
- **Applications**, voir ["Décrire une application avec HOPEX Application Design"](#), page 78.
- **Contrats de service & Catalogues**, voir ["Décrire les échanges de données"](#), page 163.
- **Processus applicatifs, Lots et IHM**, voir ["Décrire les processus applicatifs"](#), page 84.
- **Technologies**, voir le guide **HOPEX IT Architecture**.
- **Piles de technologies**, voir le guide **HOPEX IT Architecture**.
- **Implémentation UML**, voir ["A propos de l'implémentation UML"](#).
- **Traitement de lots**, voir ["Décrire les traitements de lot"](#), page 91.
- **Architectures techniques**, voir ["Décrire les services applicatifs et les micro-services"](#), page 81.

Présentation du bureau du Contributeur de conception d'application

Avec **HOPEX Application Design**, le contributeur de conception d'application dispose des mêmes volets que le Propriétaire d'application, voir "[Présentation du bureau du Propriétaire d'application](#)", page 24.

Présentation du bureau du Concepteur UML

Outre les volets proposés en standard à l'ensemble des utilisateurs du bureau de **HOPEX Application Design**, le concepteur UML dispose du volet **Ressources des applications** qui donne accès aux objets du référentiel référencés éventuellement dans le projet :

- **Processus**, voir le guide **HOPEX Business Process Analysis**.
- **Applications**, voir "[Décrire une application avec HOPEX Application Design](#)", page 78.
- **Contrats de service & Catalogues**, voir "[Décrire les échanges de données](#)", page 163.
- **Processus applicatifs, Lots et IHM**, voir "[Décrire les processus applicatifs](#)", page 84.
- **Technologies**, voir le guide **HOPEX IT Architecture**.
- **Piles de technologies**, voir le guide **HOPEX IT Architecture**.
- **Implémentation UML**, voir "[A propos de l'implémentation UML](#)".
- **Traitement de lots**, voir "[Décrire les traitements de lot](#)", page 91.
- **Architectures techniques**, voir "[Décrire les services applicatifs et les micro-services](#)", page 81.

A PROPOS DE CE GUIDE

Ce guide vous présente comment utiliser **HOPEX Application Design** pour élaborer les spécifications d'un projet IT.

Structure du guide


Le guide **HOPEX Application Design** est composé des chapitres suivants :

- "Créer un projet de conception d'application", page 31 ; décrit les fonctionnalités offertes par **HOPEX Application Design** à un administrateur fonctionnel pour préparer un projet.
- "Définir un projet AD et son périmètre applicatif", page 31 : présente comment décrire les parties prenantes sur un projet, les applications impactées ainsi que la décomposition éventuelle du projet en sous-projets.
- "Définir les objectifs et exigences d'un projet (AD)", page 43 : décrit comment spécifier les objectifs et les exigences couverts par un projet et comment les relier aux livrables du projet.
- "Décrire les spécifications fonctionnelles d'un projet", page 53, explique comment décrire les fonctionnalités attendues du projet ainsi que leur mise en oeuvre logique sur les applications du périmètre.
- "Décrire les données d'un projet", page 67, explique comment décrire les données utilisées dans le projet et leur utilisation dans les différents applications et services du projet.
- "Décrire les spécifications techniques d'un projet", page 77 : indique comment modéliser les processus de traitement informatique mis en oeuvre dans les différents cas d'utilisation d'une application ou d'un service. Il présente les interactions, qui permettent de représenter, pour la compléter ou la valider, l'architecture des services. Les interfaces reliant les services ou les opérations avec l'extérieur sont détaillées.

☛ Les rapports proposés par **HOPEX Application Design** pour assister les utilisateurs à chaque étape du projet sont présentés dans les chapitres relatifs à chacune des étapes.


Ressources complémentaires

Ce guide est complété par :

- le guide **HOPEX Common Features**, qui décrit l'interface Web et les outils spécifiques aux solutions **HOPEX**.
 *Il peut être utile de consulter ce guide pour une présentation générale de l'interface.*
- le guide **HOPEX Information Architecture**, qui décrit les modèles de données.
- des fonctions techniques plus avancées sont décrites dans le guide **HOPEX Power Studio**.
- le guide d'administration **HOPEX Power Supervisor**.

Conventions utilisées dans le guide

 Remarque sur les points qui précèdent.

 Définition des termes employés.

 Astuce qui peut faciliter la vie de l'utilisateur.

 Compatibilité avec les versions précédentes.

 **Ce qu'il faut éviter de faire.**



Remarque très importante à prendre en compte pour ne pas commettre d'erreurs durant une manipulation.

Les commandes sont présentées ainsi : **Fichier > Ouvrir**.

Les noms de produits et de modules techniques sont présentés ainsi : **HOPEX**.



Spécification d'architecture



DÉFINIR UN PROJET AD ET SON PÉRIMÈTRE APPLICATIF



HOPEX Application Design s'appuie sur une démarche méthodologique basée sur la notion de projet de spécification. Les différentes étapes du projet sont encadrées par un processus de validation soutenue par un workflow.

Dans un premier temps il convient de définir le périmètre applicatif du projet, la décomposition éventuelle en sous-projets et l'organisation des intervenants.

Les points suivants sont abordés dans ce chapitre :

- ✓ "Créer un projet de conception d'application", page 32
- ✓ "Décrire les livrables d'un projet AD", page 34,
- ✓ "Décrire l'organisation des participants au projet", page 36,
- ✓ "Décrire les sous-projets d'un projet AD", page 39,
- ✓ "Générer le rapport d'étude préliminaire", page 42

CRÉER UN PROJET DE CONCEPTION D'APPLICATION

Créer le projet

Un projet AD (Application Design) est un projet d'Architecture d'Entreprise (AE) de type "Conception d'application" dédié à la définition de spécifications logicielles. Il est associé un workflow de validation.



Un projet d'Architecture d'Entreprise est une partie d'un système dont l'étude est confiée à une même équipe, qui transforme un système, ou une partie d'un système, en vue d'atteindre un objectif donné.

Pour créer un **projet AD** :

1. A partir du volet de navigation **Accueil**, cliquez sur la tuile **Nouveau projet AD**.

Un projet d'architecture d'entreprise de type **Conception d'application** est créé. Sa fenêtre de propriétés s'ouvre dans la zone d'édition.



*A partir du volet de navigation **Environnement** > **Projets** vous pouvez accéder à l'arborescence des projets.*

2. Dans la page **Caractéristiques**, précisez le nom du projet et spécifiez éventuellement le **Détenteur**.



*Pour plus de détails sur la gestion des bibliothèques, voir le guide **HOPEX Common Features**, "Entreprises et bibliothèques".*

3. Spécifiez également les intervenants du projet. Voir "[Décrire l'organisation des participants au projet](#)", page 36.

Sous la page **Assignation de l'environnement de travail**, vous pouvez visualiser l'environnement de travail créé pour le projet.

Assigner un environnement de travail au projet

La gestion d'un projet de conception d'application comprend différentes étapes : collecte des besoins, définition du périmètre, etc.


Dans **HOPEX Application Design**, ces étapes sont décrites dans un **environnement de travail** de type **Conception d'application**.

Une fois l'environnement de travail créé sur un projet, vous pouvez l'afficher dans l'interface. Un volet de navigation **Projet en cours** affiche les différentes étapes du projet sélectionné, avec les outils nécessaires à chacune d'elles.

Créer un environnement de travail sur un projet

Pour définir un environnement de travail sur un projet :

1. Dans la page d'accueil, cliquez sur **Tous les projets AE**.

2. Sélectionnez le projet en question pour afficher ses propriétés.
 Cliquez sur le bouton **Propriétés** de la zone d'édition si les propriétés ne s'affichent pas.
3. Sélectionnez la page **Assignment de l'environnement de travail**.
4. Cliquez sur **Nouveau**.
5. Renommez si besoin le nouvel environnement et sélectionnez le type "Application Design".
6. Cliquez sur **OK**.

Par défaut les étapes de l'environnement de travail sont visibles par tous les utilisateurs. Vous pouvez définir plus précisément les participants au projet, qui pourront valider chaque étape du projet à travers le workflow de validation du projet. Voir ["Décrire l'organisation des participants au projet"](#), page 36.

Afficher l'environnement de travail d'un projet

Pour afficher l'environnement de travail d'un projet :

1. Cliquez sur **Menu principal**.
2. Sélectionnez **Changer d'environnement de travail** et choisissez le projet sur lequel vous souhaitez travailler.

Voir aussi : ["Le volet Projet en cours"](#), page 22.

DÉCRIRE LES LIVRABLES D'UN PROJET AD

Décrire les livrables d'un projet (AD) consiste à définir le périmètre du projet et la liste des applications et services qui seront ajoutés ou modifiés.



Un Livrable de projet définit le résultat d'un projet et son impact ou sa contribution sur le paysage des solutions d'architecture de l'entreprise.

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Décrire le périmètre applicatif du projet

Exemple de périmètre applicatif

L'application "Achat de matériel" est la principale application impactée par le projet, en particulier pour l'amélioration de son portail d'accès.

L'application "Gestion des stocks" est utilisée par l'application "Achat de matériel" pour consulter et mettre à jour les informations concernant l'état des stocks et, en particulier, celui des pièces détachées.

L'application "Gestion des achats" utilise la base de données "Catalogue" pour passer une commande d'achat.

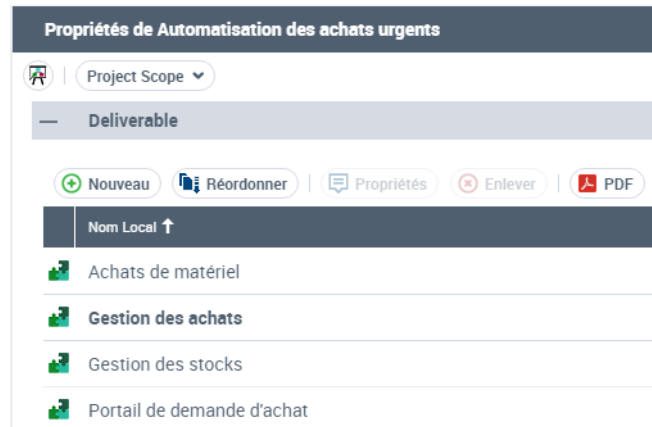
L'application "Gestion Entretien et Réparation" et l'application "Gestion des pièces de rechange" vont également utiliser le portail de l'application "Achats de matériel".

Décrire les applications du projet

Pour ajouter une application existante à la liste des applications comprises dans le périmètre du projet en cours :

1. A partir du volet de navigation **Projet en cours > Définition de projet**, sélectionnez **Périmètre de livrables de projet**.
La liste des applications impactées par le projet en cours apparaît dans la section **Applications spécifiées**.
2. Cliquez sur le bouton **Nouveau**.
Une fenêtre d'ajout d'**Application** apparaît.

3. A partir du champ **Nom**, sélectionnez l'application que vous souhaitez ajouter et cliquez sur **OK**.
La nouvelle application apparaît dans la liste des applications impactées par le projet.



Décrire les résultats produits par le projet

Accéder à la liste des résultats produits par le projet en cours

Pour accéder à la liste des *résultats* produits par le projet en cours :

1. A partir du volet de navigation **Projet en cours > Définition du projet en cours**, sélectionnez **Vue hiérarchique du projet**.
La liste des livrables attendus du projet en cours apparaît dans le dossier **Livrables produits**.

Les applications qui constituent le périmètre du projet sont automatiquement ajoutées aux résultats du projet.


Ajouter un nouveau résultat de projet

Pour spécifier qu'un objet existant est un **résultat** pour le projet en cours :

1. A partir du volet de navigation **Projet en cours > Définition du projet en cours**, sélectionnez **Périmètre de livrables de projet**.
La liste des résultats attendus du projet en cours apparaît dans la section **Autres objets**.
2. Cliquez sur le bouton **Nouveau**.
Une fenêtre d'ajout de **Résultat de projet** apparaît.
3. A partir du champ **Type d'objet**, sélectionnez le type du résultat que vous souhaitez ajouter.
4. A partir du champ **Nom**, sélectionnez l'objet que vous souhaitez déclarer comme résultat et cliquez sur **OK**.
L'objet apparaît dans la liste des résultats du projet.


DÉCRIRE L'ORGANISATION DES PARTICIPANTS AU PROJET


Les participants à un projet de spécification sont des *personnes (système)* regroupées dans différentes *organisations*.

 Une *personne (Système)* représente une personne de l'entreprise. Cette personne est peut être associée à un login et un rôle (ou un profil selon le mode de connexion). Le login donne accès à l'application HOPEX. Le rôle (ou le profil) définit les droits d'accès aux référentiels et aux fonctionnalités du produit. Une *personne système*, si elle est associée à un login, dispose, dans chaque référentiel, d'un bureau qui lui est propre auquel elle peut se connecter à partir de n'importe quel poste d'un environnement donné.

 Une *organisation d'Architecture d'Entreprise* est un organe de gouvernance regroupant différentes personnes ou organisations.

Une même personne peut intervenir dans différentes organisations avec des *rôles métier* différents pour chacune d'entre elles.

 Un *rôle métier* permet d'assigner une tâche à une personne (ex. : *décrire une application*) et si besoin pour une localisation spécifique (ex. : *site de paris*). Des rôles métier sont assignés à des personnes et une personne peut avoir plusieurs rôles métier.

 Pour plus de détails sur les rôles métier livrées avec **HOPEX Application Design**, voir "*Les rôles métier de HOPEX Application Design*", page 4.

Une **Affiliation de personne** permet de relier une *personne (système)* à un *rôle métier* dans le contexte d'une organisation définie pour un projet.

De la même manière, une **Affiliation d'organisation** permet de définir une hiérarchie organisationnelle en reliant une organisation à une autre dans le contexte d'un projet.

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "*Le volet Projet en cours*", page 22.

Accéder à l'organisation des participants à un projet

Pour accéder à la liste des *organisations AE* d'un projet de spécification :

1. Ouvrez la page de propriétés **Caractéristiques** du projet qui vous intéresse.
2. Dépliez la section **Organisation**.
La liste des *organisations AE* de plus haut niveau apparaît.

Pour accéder à la description d'une organisation du projet en cours :

1. A partir du volet **Projet en cours > Définition du projet en cours**, cliquez sur la tuile **Parties prenantes**.
La liste des **organisations AE** de plus haut niveau reliées au projet en cours apparaît.
2. Cliquez sur l'organisation qui vous intéresse.
La liste des **Affiliations de personne** ainsi que la liste des **Affiliation d'organisation** qui constituent l'organisation décrite apparaissent.

Créer une organisation

Pour créer une **Organisation** pour le projet en cours :

1. A partir du volet **Projet en cours > Définition du projet en cours**, cliquez sur la tuile **Parties prenantes**.
La liste des **organisations AE** de plus haut niveau reliées au projet en cours apparaît.
2. Cliquez sur le bouton **Nouveau**.
Une fenêtre de création d'**Organisation** apparaît.
3. Spécifiez le nom de l'organisation que vous souhaitez créer et cliquez sur **OK**.
La nouvelle organisation apparaît dans la liste des organisations du projet.

Relier une personne à une organisation

Pour créer une **Affiliation de personne** dans une organisation reliée au projet en cours :

1. A partir du volet de navigation **Projet en cours > Définition du projet en cours**, cliquez sur la tuile **Parties prenantes**.
La liste des **organisations AE** de plus haut niveau reliées au projet en cours apparaît.
2. Dépliez la section **Organisation** et sélectionnez l'organisation pour laquelle les personnes interviennent.
La liste des **Affiliations de personne** ainsi que la liste des **Affiliation d'organisation** qui constituent l'affiliation décrite apparaissent.
3. Dans la section **Affiliation de personne**, cliquez sur le bouton **Nouveau**.
Une fenêtre de création d'**Affiliation de personne** apparaît.
4. Dans le champ **Personne membre**, cliquez sur **Relier Personne (système)**.
Une fenêtre de connexion de personne système s'ouvre.
5. Recherchez et sélectionnez la personne associée à l'utilisateur qui vous intéresse et cliquez sur **Relier**.
6. Dans le champ **Rôle dans l'organisation**, cliquez sur **Relier Rôle métier**.
Une fenêtre de connexion s'ouvre.
7. Recherchez et sélectionnez le rôle métier qui vous intéresse et cliquez sur **Relier**.

8. Cliquez sur **Relier**.

La nouvelle affiliation de personne apparaît dans la liste avec le nom de l'utilisateur et son rôle dans l'organisation décrite.

Réutiliser une organisation existante dans un projet

Pour réutiliser une **Organisation** à partir d'un projet :

1. Ouvrez la page de propriétés **Caractéristiques** du **projet** qui vous intéresse.
2. Dépliez la section **Organisation** et sélectionnez l'organisation pour laquelle les personnes interviennent.
La liste des **Affiliations de personne** ainsi que la liste des **Affiliation d'organisation** qui constituent l'affiliation décrite apparaissent.
3. Dans la section **Affiliation d'organisation**, cliquez sur le bouton **Nouveau**.
Une fenêtre de création d'**Affiliation d'organisation** apparaît.
4. Dans la section **Mode de création**, cliquez sur **Réutiliser une affiliation d'organisation**.
Une section **Sous-organisation** apparaît avec la liste des organisations existantes.
5. Sélectionnez l'organisation qui vous intéresse et cliquez sur **OK**.
La nouvelle affiliation d'organisation apparaît dans la liste.

DÉCRIRE LES SOUS-PROJETS D'UN PROJET AD

Un projet AD peut être décomposé en plusieurs sous-projets.

Il est également possible de décrire des liens de dépendance entre les différents sous-projets.

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Accéder à la décomposition d'un projet en sous-projets

Pour accéder à la liste des sous-projets d'un projet AD :

1. Ouvrez la page de propriétés **Sous-projets** du **projet** qui vous intéresse.
2. Dépliez la section **Sous-projets**.
La liste des sous-projets de plus haut niveau apparaît.

☛ La page de propriétés **Sous-projets** d'un projet permet d'accéder à la liste des sous-projets du projet décrit.

Relier un sous-projet au projet en cours

Les projets sont créés par un **Administrateur fonctionnel de conception d'application** ou par un **Concepteur d'application**.

☛ Pour plus de détails sur ces profils, voir "[L'administrateur fonctionnel de conception d'application](#)", page 3 et "[Le concepteur d'application](#)", page 3.

☛ Pour plus de détails sur la création de projet avec **HOPEX Application Design**, voir "[Créer un projet de conception d'application](#)", page 32.

Pour relier un projet existant au projet en cours :

1. A partir du volet **Projet en cours > Définition de projet**, cliquez sur **Sous-projets**.
La liste des sous-projets de plus haut niveau reliés au projet en cours apparaît.
2. Dans la section **Sous-projets**, cliquez sur le bouton **Relier**.
Une fenêtre de connexion de projet AE s'ouvre.

3. Recherchez et sélectionnez le projet qui vous intéresse et cliquez sur **Relier**.

Le projet apparaît dans la liste des sous-projets du projet décrit.

☛ Pour accéder à la hiérarchie des projet à partir du volet **Projet en cours > Définition de projet**, cliquez sur **Hiérarchie des projets courants** et dépliez les dossier **Sous-Projet** des projets qui apparaissent dans l'arbre.

Décrire les dépendances du projet en cours

Vous pouvez spécifier la liste des projets dont le projet en cours dépend. Pour chacun de ces projets, vous pouvez également préciser quel **résultat** du projet en cours est impacté par cette dépendance.

Créer une dépendance de projet

Pour spécifier que le projet en cours dépend d'un autre projet :

1. A partir du volet **Projet en cours > Définition de projet**, cliquez sur **Sous-projets**.
La liste des sous-projets de plus haut niveau reliés au projet en cours apparaît.
2. Dans la section **Dépendances du projet**, cliquez sur le bouton **Nouveau**.
Une fenêtre de création de **Dépendance de projet** s'ouvre.
3. A partir du champ **Projet attendu**, recherchez et sélectionnez le projet qui vous intéresse et cliquez sur **Ajouter**.
La **dépendance de projet** est créée.

Spécifier le résultat d'une dépendance de projet

Pour spécifier les livrables de projet impactés par une dépendance de projet :

1. Ouvrez la page de propriétés **Caractéristiques** de la dépendance qui vous intéresse.
2. Dans la section **Livrables concernés**, cliquez sur le bouton **Relier**.
Une fenêtre de connexion de **Résultat de projet** s'ouvre.
3. Sélectionnez le résultat de projet qui vous intéresse et cliquez sur **Relier**.
Le résultat apparaît dans la liste des livrables concernés.

Spécifier les exigences d'une dépendance de projet



Une exigence est un besoin ou une attente formulés explicitement, imposés comme une contrainte à respecter dans le cadre d'un projet de certification, d'organisation ou de modification du système d'information d'une entreprise.

☛ Pour plus de détails sur les exigences d'un projet de spécification, voir "[Créer un projet](#)", page 31.

Pour spécifier les exigences de projet imposées par une dépendance de projet :

1. Ouvrez la page de propriétés **Objectifs et exigences** de la dépendance qui vous intéresse.

2. Sélectionnez le dossier **Exigences imposées**, cliquez sur le bouton **Relier**.
Une fenêtre de connexion d'**Exigence** s'ouvre.
3. Sélectionnez l'exigence qui vous intéresse et cliquez sur **Relier**.
L'exigence apparaît dans la liste des exigences imposées.


GÉNÉRER LE RAPPORT D'ÉTUDE PRÉLIMINAIRE

Le rapport d'étude préliminaire offre dans un document Word une description du projet en cours. Il présente les éléments suivants :

- les livrables du projet
- les parties prenantes
- les objectifs et exigences du projet, ainsi que les exigences associées à ses livrables
- les processus concernés par le projet
- les dépendances du projet
- les étapes du workflow du projet

Pour générer un rapport d'étude préliminaire sur un projet en cours :

1. A partir du volet **Projet en cours > Définition de projet**, cliquez sur **Rapport d'étude préliminaire**.

 Vous devez avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22

La liste des rapports MS Word reliés au projet en cours apparaît.

2. Dans la zone d'édition cliquez sur le bouton **Nouveau**.
Une fenêtre de création de rapport MS Word s'ouvre.
3. Spécifiez le nom du rapport.
4. Dans le champ **Rapport type (MS Word)**, sélectionnez **Application Design - Etude préalable**.
5. Cliquez sur **OK**.
Le rapport apparaît dans la liste.
6. Cliquez sur l'icône du nouveau rapport puis sur **Documentation > Ouvrir**.

DÉFINIR LES OBJECTIFS ET EXIGENCES D'UN PROJET



Dans un projet de conception d'application, une fois le périmètre décrit, la seconde étape vise à représenter les objectifs et les exigences du projet.


Les exigences sont ensuite déclinées sur les composants concernés, qu'ils soient techniques ou organisationnels.


Les points suivants sont abordés dans ce chapitre :


- ✓ ["Recenser les objectifs et les exigences d'un projet", page 44,](#)
- ✓ ["Utiliser un diagramme d'objectifs et d'exigences", page 47,](#)
- ✓ ["Générer le rapport d'analyse des exigences", page 52.](#)

RECENSER LES OBJECTIFS ET LES EXIGENCES D'UN PROJET

Le premier travail à réaliser, après la création et l'initialisation d'un **projet**, est de recenser les **objectifs** visés ainsi que les **exigences** imposées afin de les relier aux principaux composants organisationnels et techniques qui constituent le périmètre du projet.

 *Un projet d'Architecture d'Entreprise est une partie d'un système dont l'étude est confiée à une même équipe, qui transforme un système, ou une partie d'un système, en vue d'atteindre un objectif donné.*

 *Un objectif est un but que l'on cherche à atteindre ou la cible visée par un processus ou une opération. Il permet de mettre en évidence les points que l'on veut améliorer pour ce processus ou cette opération.*

 *Une exigence est un besoin ou une attente formulés explicitement, imposés comme une contrainte à respecter dans le cadre d'un projet de certification, d'organisation ou de modification du système d'information d'une entreprise.*

Ce paragraphe décrit :

- "Gérer la liste des objectifs d'un projet", page 44,
- "Gérer la liste des exigences d'un projet", page 45.

Gérer la liste des objectifs d'un projet

Accéder aux objectifs d'un projet

Pour accéder à la liste des **objectifs** d'un projet de spécification :

1. Ouvrez la page de propriétés **Analyse des besoins** du projet qui vous intéresse.
2. Dépliez la section **Objectifs**.
Notez que deux listes d'**objectifs** vous sont proposées :
 - les **objectifs assignés**, qui représentent les objectifs prévus dans le cadre du projet,
 - les **objectifs réalisés**, qui représentent les objectifs déjà atteints.

Créer un nouvel objectif pour le projet en cours

Pour créer un **objectif** dans le contexte du projet en cours :

1. A partir du volet **Projet en cours > Analyse des besoins**, sélectionnez cliquez sur **Objectifs**.
Une liste des **objectifs** reliés au projet en cours apparaît.
2. Sélectionnez la liste des **objectifs assignés** et cliquez sur le bouton **Nouveau**.
Une fenêtre de création d'un objectif apparaît.

3. Saisissez le nom de l'objectif et cliquez sur **OK**.
L'objectif apparaît dans la liste.

Consulter les propriétés d'un objectif du projet en cours

Pour consulter les propriétés d'un **objectif** du projet en cours :

1. A partir du volet **Projet en cours > Analyse des besoins**, cliquez sur **Objectifs**.
Une liste des **objectifs** reliés au projet en cours apparaît.
2. Ouvrez les pages de propriétés de l'objectif qui vous intéresse.

La page **Caractéristiques** des propriétés d'un **objectif** permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de la création de l'objectif, il s'agit du projet en cours.
- le texte de sa **Description**,
- le **Type d'objectif** : Qualitatif, Quantitatif ou Autre.
- la liste de ses **sous-objectifs**.

Avec **HOPEX Application Design** un objectif est décrit par les pages suivantes :

- le page **Portée** permet d'accéder à la liste des composants contraints par l'objectif.
- la pages **Contributeurs** permet d'accéder à la liste des composants qui contribuent au fait que l'objectif soit atteint.

✎ Pour plus de détails la manière de relier un objectif à des éléments du périmètre d'un projet, voir "[Relier les composants techniques et organisationnels aux exigences](#)", page 50.

- la page **Exigences** permet d'accéder à la liste des exigences reliées à l'objectif décrit.

📖 Une exigence est un besoin ou une attente formulés explicitement, imposés comme une contrainte à respecter dans le cadre d'un projet de certification, d'organisation ou de modification du système d'information d'une entreprise.

✎ Pour plus de détails sur la manière de relier un objectif exigence à des , voir "[Créer un diagramme d'objectifs et d'exigences d'un projet](#)", page 49.

Gérer la liste des exigences d'un projet

Accéder aux exigences d'un projet

Pour accéder à la liste des **exigences** d'un projet de spécification :

1. Ouvrez la page de propriétés **Analyse des besoins** du projet qui vous intéresse.
2. Dépliez la section **Exigences**.
La liste des **exigences** apparaît.

Créer une nouvelle exigence pour le projet en cours

Pour créer une **exigence** dans le contexte du projet en cours :

1. A partir du volet **Projet en cours > Analyse des besoins**, cliquez sur **Exigences**.
La liste des **exigences** reliées au projet en cours apparaît.
2. Cliquez sur le bouton **Nouveau**.
Une fenêtre de création d'une exigence apparaît.
3. Saisissez le nom de l'exigence et cliquez sur **OK**.
L'exigence apparaît dans la liste.

Consulter les propriétés d'une exigence du projet en cours

Pour consulter les propriétés d'une **exigence** du projet en cours :

1. A partir du volet **Projet en cours > Analyse des besoins**, cliquez sur **Exigences**.
La liste des **exigences** reliées au projet en cours apparaît.
2. Ouvrez les pages de propriétés de l'exigence qui vous intéresse.

La page **Caractéristiques** des propriétés d'une **exigence** permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de la création de l'exigence, il s'agit du projet en cours.
- le texte de sa **Description**,
- sa **Nature d'exigence** : Technique, Système, Organisationnelle, etc.
- la liste de ses **sous-exigences**.

Avec **HOPEX Application Design** une exigence est décrite par les pages suivantes :

- les pages **Portée** et **Contributeurs** qui permettent d'accéder à la liste des composants qui sont, respectivement, contraints par l'exigence ou qui contribuent au fait qu'elle soit satisfaite.
 - ☛ Pour plus de détails la manière de relier une exigence à des éléments du périmètre d'un projet, voir "[Relier une application à une exigence](#)", page 51.
- la page **Objectifs** permet d'accéder à la liste des objectifs reliés à l'exigence décrite.




Un objectif est un but que l'on cherche à atteindre ou la cible visée par un processus ou une opération. Il permet de mettre en évidence les points que l'on veut améliorer pour ce processus ou cette opération.




Pour plus de détails sur la manière de relier une exigence à des objectifs, voir "[Relier associer un objectif à une exigence dans un diagramme d'objectifs et d'exigences](#)", page 49.

UTILISER UN DIAGRAMME D'OBJECTIFS ET D'EXIGENCES

Avec **HOPEX Application Design**, vous pouvez construire votre diagramme d'*objectifs* et d'*exigences* à partir d'un projet de spécification.

 *Un objectif est un but que l'on cherche à atteindre ou la cible visée par un processus ou une opération. Il permet de mettre en évidence les points que l'on veut améliorer pour ce processus ou cette opération.*

 *Une exigence est un besoin ou une attente formulés explicitement, imposés comme une contrainte à respecter dans le cadre d'un projet de certification, d'organisation ou de modification du système d'information d'une entreprise.*

Ce paragraphe décrit :

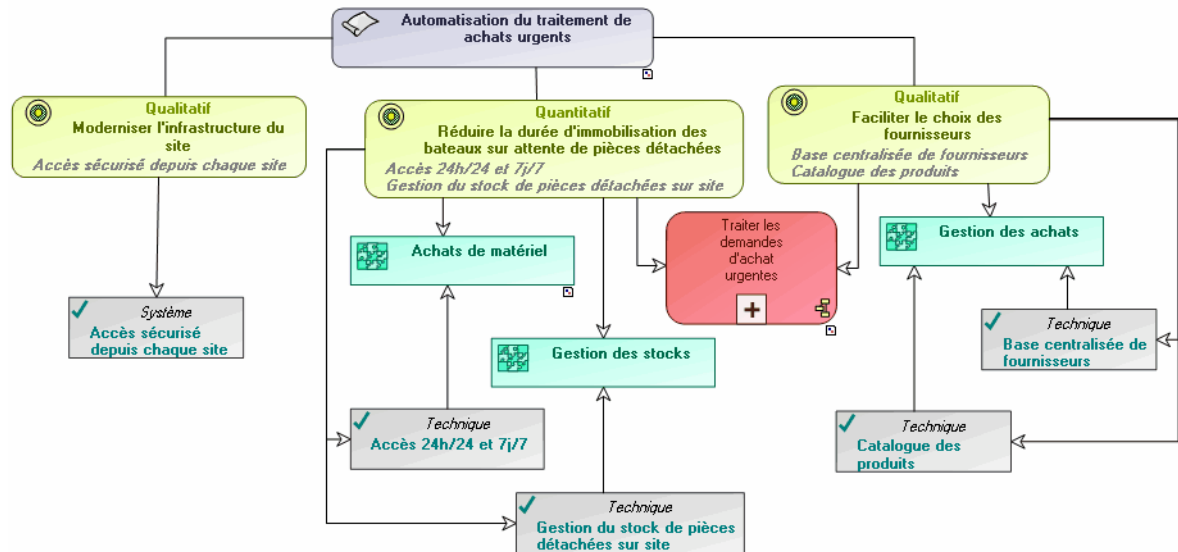
- ["Consulter un diagramme d'objectifs et d'exigences"](#), page 47,
- ["Créer un diagramme d'objectifs et d'exigences d'un projet"](#), page 49,
- ["Relier les composants techniques et organisationnels aux exigences"](#), page 50.

Consulter un diagramme d'objectifs et d'exigences

Pour consulter le diagramme d'objectifs et d'exigences d'un projet :

1. Ouvrez le menu contextuel du projet.

- Faites un double-clic sur **Diagramme d'objectifs et d'exigences du projet**.
Le diagramme s'ouvre.



Dans cet exemple, le projet répond à un objectif quantitatif de réduction de la durée d'immobilisation des bateaux et à des objectifs qualitatifs d'amélioration du système de communication, d'une part, et de gestion des fournisseurs, d'autre part.

Ces objectifs sont liés à des exigences techniques ou système telles que l'accès 24h sur 24 et 7 jours sur 7 à l'application de demande d'achats urgents.

Le processus de traitement des achats urgents doit respecter les objectifs de l'entreprise au niveau de la réduction du temps d'immobilisation des bateaux et de la facilité d'accès aux fournisseurs et aux produits.

Par ailleurs, l'application d'achat de matériel doit être accessible 24h sur 24 et 7 jours sur 7. L'application de gestion des stocks doit intégrer les exigences de mise en place du catalogue de produits et de la base de fournisseurs. Enfin, l'application de gestion des stocks doit intégrer l'exigence de gestion des stocks locaux de pièces détachées.

☛ Pour faire apparaître les applications ou les processus organisationnels dans un diagramme d'objectifs et d'exigences, activez les vues correspondantes à partir du menu **Affichage > Vues et détails**.

☛ Les processus organisationnels sont disponibles avec le produit **HOPEX Business Process Analysis**.

Créer un diagramme d'objectifs et d'exigences d'un projet

Créer un diagramme d'objectifs et d'exigences d'un projet

Pour créer un nouveau diagramme d'objectifs et d'exigences de projet :

1. Cliquez avec le bouton droit sur votre projet et sélectionnez **Nouveau > Diagramme d'objectifs et d'exigences d'un projet**.
Un diagramme vide apparaît.

☛ Pour plus de d'informations sur les diagrammes d'objectifs et d'exigences, voir le guide **HOPEX Common Features**, "Objectifs et exigences".

2. A partir de l'arbre de navigation, cliquez sur le projet décrit pour le faire glisser dans le diagramme.

Ajouter les objectifs dans le diagramme d'objectifs et d'exigences du projet

Pour ajouter un *objectif* :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Objectif** puis cliquez dans le diagramme.
La fenêtre d'ajout d'un objectif apparaît.
2. A partir du champ **Nom**, sélectionnez **Lister**.
3. Sélectionnez l'objectif qui vous intéresse et cliquez sur **OK**.
L'objectif apparaît dans le diagramme ainsi que dans l'arbre de navigation du projet.



☛ Si le projet décrit est positionné dans le diagramme, un lien apparaît automatiquement entre le projet et l'objectif.

Ajouter les exigences dans le diagramme d'objectifs et d'exigences du projet

Pour ajouter une *exigence* :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Exigence** puis cliquez dans le diagramme.
La fenêtre d'ajout d'une exigence apparaît.
2. A partir du champ **Nom**, sélectionnez **Lister**.
3. Sélectionnez l'exigence qui vous intéresse et cliquez sur **OK**.
L'exigence apparaît dans le diagramme.

☛ Si le projet décrit est positionné dans le diagramme, un lien apparaît automatiquement entre le projet et l'exigence.

Relier associer un objectif à une exigence dans un diagramme d'objectifs et d'exigences

Pour spécifier l'objectif associé à une exigence à partir d'un diagramme d'objectif et d'exigence d'un projet :

1. Ouvrez le diagramme

2. Cliquez sur la bobine.
3. Cliquez sur l'objectif et, en maintenant le bouton de la souris enfoncé, tirez un lien vers l'exigence associée.
Un lien orienté de l'objectif vers l'exigence apparaît dans le diagramme.
Le nom de l'exigence apparaît dans l'icône de l'objectif.

☛ Le lien peut aussi être créé à partir de l'exigence vers l'objectif.

Les pages de propriétés de l'objectif et de l'exigence sont alors mises à jour.

☛ Pour plus de détails sur les propriétés d'une exigence ou d'un objectif, voir "[Consulter les propriétés d'un objectif du projet en cours](#)", page 45 et "[Consulter les propriétés d'une exigence du projet en cours](#)", page 46.

Relier les composants techniques et organisationnels aux exigences

Rappel du principe

Les composants techniques et organisationnels sur lesquels portent les objectifs et les exigences peuvent être précisés.


Par exemple, l'application "Gestion des achats" doit respecter l'exigence portant sur la mise en place d'un catalogue de produits ainsi que l'exigence de mise en place d'une base centralisée de fournisseurs.

Ces liens peuvent être créés graphiquement dans le diagramme d'objectifs et d'exigences du projet.

Ajouter une application dans un diagramme d'objectifs et d'exigences

Pour relier une application à une exigence à partir d'un diagramme d'objectifs et d'exigences, il faut d'abord faire apparaître les applications dans le diagramme puisqu'elles ne font pas partie des vues par défaut.

Pour visualiser les applications dans un diagramme d'objectifs et d'exigences d'un projet :

1. Cliquez sur le bouton **Vues et détails**  de la barre d'outils du diagramme.
2. Cochez la case **Applications, Services, Bases de données**.
3. Cliquez sur **OK**.
Les boutons correspondants apparaissent dans la barre d'objets.

Pour poser une application dans un diagramme d'objectifs et d'exigences :

1. Cliquez sur le bouton **Application** de la barre d'objets.
2. Cliquez dans le diagramme.
La fenêtre d'ajout d'une application s'ouvre.
3. Cliquez sur la flèche à droite du champ **Nom** et sélectionnez **Lister** dans la liste déroulante.
La liste des applications accessibles à partir de la bibliothèque courante apparaît.


4. Sélectionnez l'application de votre choix.
5. Cliquez sur le bouton **Ajouter**.
L'application apparaît dans le diagramme.

Relier une application à une exigence


Relier une application à une exigence peut avoir deux significations non exclusives :

- Soit l'application **contribue** à la satisfaction d'une exigence,
- Soit l'application est **contrainte** par l'exigence.

Pour spécifier graphiquement qu'une application contribue à la satisfaction d'une exigence :

1. Cliquez sur la bobine  de la barre d'objets.
2. Cliquez sur l'application et, en maintenant le bouton de la souris enfoncé, tirez un lien vers l'exigence.
Le lien orienté apparaît en pointillé de l'application vers l'exigence.

Pour spécifier graphiquement qu'une application est contrainte par la satisfaction d'une exigence :

1. Cliquez sur la bobine  de la barre d'objets.
2. Cliquez sur l'exigence et, en maintenant le bouton de la souris enfoncé, tirez un lien vers l'application.
3. Lâchez le bouton.
Le lien orienté apparaît de l'exigence vers l'application.

Les pages de propriétés de l'exigence et de l'objectif associé sont alors mis à jour.

➡ Pour plus de détails sur les propriétés d'une exigence ou d'un objectif, voir ["Consulter les propriétés d'un objectif du projet en cours", page 45](#) et ["Consulter les propriétés d'une exigence du projet en cours", page 46](#).

GÉNÉRER LE RAPPORT D'ANALYSE DES EXIGENCES

Ce rapport type décrit le projet, ses objectifs et ses exigences ainsi que la liste des processus métiers, organisationnels et fonctionnels contenus dans le projet et liés à des exigences.

➡ Pour plus de détails sur la descriptions des objectifs et des exigences d'un projet voir : ["Recenser les objectifs et les exigences d'un projet", page 44.](#)

Le rapport généré présente :

- le diagramme d'objectifs et d'exigences du projet.
- le diagramme des processus détenus dans le projet.
- pour chaque processus détenu dans le projet, le détail des exigences liées à ces composants.

<div>Print Shopping cart</div> <div>Complete order</div>	Information on shopping cart item lines			Order payment	Order validation
	<div> <div>Object</div> <div>Type</div> <div>Link</div> </div>				
	1	Print Shopping cart	Operation	Imposed Requirement	
	2	Information on shopping cart item lines	Requirement		
				+	+

DÉCRIRE LES SPÉCIFICATIONS FONCTIONNELLES D'UN PROJET



Après avoir décrit le périmètre du projet, ses objectifs et ses exigences, il est nécessaire de spécifier les fonctionnalités attendues des principaux composants du projet.

A l'issue de cette étape de spécifications fonctionnelles, la direction de l'entreprise peut décider de la poursuite, ou non, du projet.

- ✓ ["Décrire les fonctionnalités d'un projet", page 54,](#)
- ✓ ["Décrire un environnement d'application", page 58,](#)
- ✓ ["Décrire un cas d'utilisation d'un projet", page 62,](#)
- ✓ ["Générer le rapport d'analyse fonctionnelle", page 65.](#)

DÉCRIRE LES FONCTIONNALITÉS D'UN PROJET

La solution **HOPEX Application Design** distingue les *fonctionnalités techniques* des autres *fonctionnalités*.



Une fonctionnalité est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'une fonction métier ou d'une organisation.



Une fonctionnalité technique est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'un élément technique ou d'une application.

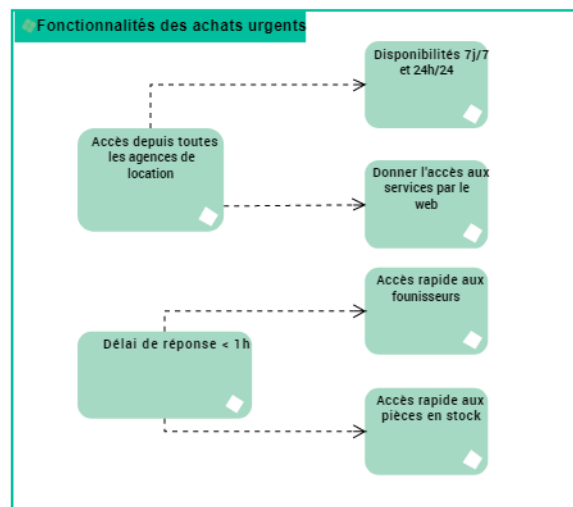
Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Décrire la carte des fonctionnalités

Une carte de fonctionnalités est un assemblage de fonctionnalités avec leurs dépendances qui, conjointement, définissent le périmètre d'une architecture matérielle ou logicielle.

Dans notre exemple, une carte de fonctionnalités se présente sous la forme du diagramme suivant.



Les fonctionnalités attendues du projet de traitement des achats urgents sont relatives au temps de réponse et à la

disponibilité de l'application depuis les agences de locations. Les fonctionnalités principales sont déclinées en sous-fonctionnalités.

Accéder à la liste des cartes de fonctionnalités

Pour accéder à la liste des cartes de fonctionnalités :

1. Cliquez sur le menu de navigation puis sur **Projets en cours**.
2. Dans le volet de navigation sélectionnez **Conception fonctionnalités et règles**.
3. Cliquez sur le dossier **Liste des cartes des fonctionnalités** et sélectionnez l'onglet **Cartes de fonctionnalités**.

La liste des cartes de fonctionnalités apparaît dans la zone d'édition.

Créer une carte de fonctionnalités

Pour créer une carte des fonctionnalités :

1. Sélectionnez **Projets en cours > Conception fonctionnalités et règles** dans le menu de navigation.
2. Cliquez sur **Liste des cartes des fonctionnalités** et sélectionnez l'onglet **Cartes de fonctionnalités**.
3. Cliquez sur le bouton **Nouveau**.
Une fenêtre de création de **Carte de fonctionnalités** apparaît.
4. Spécifiez le **Nom** de la nouvelle carte de fonctionnalités.
5. Cliquez sur **OK**.

La nouvelle carte de fonctionnalités apparaît dans la liste.

Décrire une carte de fonctionnalités

Une carte de fonctionnalités est décrite par un diagramme.

Pour créer un diagramme de carte de fonctionnalités :

1. Faites un clic droit sur la carte de fonctionnalités qui vous intéresse et cliquez sur **Nouveau > Carte de fonctionnalités**.

Le diagramme s'ouvre dans la fenêtre d'édition. Le cadre de la carte de fonctionnalités décrite apparaît dans le diagramme.

Pour ajouter une sous-fonctionnalité dans un diagramme de fonctionnalités, voir "[Créer une sous-fonctionnalité dans une carte de fonctionnalités](#)", page 55.

Pour définir les dépendances des sous-fonctionnalités, voir "[Décrire les dépendances entre les fonctionnalités à partir de la carte de fonctionnalités](#)", page 56.

Créer une sous-fonctionnalité dans une carte de fonctionnalités

Les composants représentés dans une carte des fonctionnalités sont des *fonctionnalités*.

Pour ajouter une nouvelle sous-fonctionnalité dans le diagramme d'une carte de fonctionnalités :

1. Dans la barre d'insertion du diagramme, cliquez sur le bouton **Sous fonctionnalité**.

2. Cliquez dans le cadre de la carte des fonctionnalités.
Une fenêtre d'ajout d'une fonctionnalité s'ouvre.
3. Saisissez le nom de la fonctionnalité que vous souhaitez créer.
4. Cliquez sur **OK**.
La nouvelle fonctionnalité apparaît dans le diagramme.

Décrire les dépendances entre les fonctionnalités à partir de la carte de fonctionnalités

Un lien de dépendance entre une sous-fonctionnalité et une autre permet de spécifier les éléments sur lesquels repose cette dépendance.

Par exemple, la fonctionnalité relative au temps de réponse dépend du temps d'accès à la liste des fournisseurs et au temps d'accès à la disponibilité des pièces en stock.

Pour créer des liens de dépendance entre deux fonctionnalités dans le diagramme de carte des fonctionnalités :

1. Dans la barre d'insertion, cliquez sur le bouton **Dépendance de fonctionnalité**.
2. Cliquez sur la fonctionnalité et, en maintenant le bouton gauche de la souris enfoncé, déplacez le curseur sur la fonctionnalité utilisée.
3. Relâchez le bouton de la souris.
Le lien apparaît dans le diagramme.

☛ Une même fonctionnalité peut avoir plusieurs dépendances au sein d'une même carte de fonctionnalités.

Les propriétés d'une carte de fonctionnalités

La page **Caractéristiques** des propriétés d'une carte de fonctionnalités permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de la carte de fonctionnalités, il s'agit du projet courant,
- son **Nom**,
- le texte de sa **Description**.

Avec **HOPEX Application Design** une carte de fonctionnalités est décrite par les pages suivantes :

- la page **Structure** qui permet de spécifier la liste des composants de fonctionnalité détenus et les dépendances entre eux.

☛ Pour plus de détails sur les composants d'une carte de fonctionnalités, voir ["Créer une sous-fonctionnalité dans une carte de fonctionnalités"](#), page 55 et ["Décrire les dépendances entre les fonctionnalités à partir de la carte de fonctionnalités"](#), page 56.

Décrire une fonctionnalité



Une fonctionnalité est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'une fonction métier ou d'une organisation.

Accéder à la liste des fonctionnalités

Pour accéder à la liste des cartes de fonctionnalités à partir du volet de navigation **Projets en cours** :


1. Sélectionnez **Spécification des fonctionnalités et des règles** dans le menu de navigation.
2. Cliquez sur le dossier **Liste des cartes des fonctionnalités** et sélectionnez l'onglet **Fonctionnalités**.
La liste des fonctionnalités apparaît dans la zone d'édition.

Les propriétés d'une fonctionnalité

La page **Caractéristiques** des propriétés d'une fonctionnalité permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de la carte de fonctionnalités, il s'agit du projet courant,
- son **Nom**,
- le texte de sa **Description**.

Décrire une fonctionnalité technique

 Une fonctionnalité technique est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'un élément technique ou d'une application.

Avec **HOPEX Application Design**, l'utilisation des fonctionnalités techniques et des cartes de fonctionnalités techniques est identiques à celui des fonctionnalités et des cartes de fonctionnalités.

☛ Pour plus de détails sur le fonctionnement des cartes de fonctionnalités et des fonctionnalités, voir "[Décrire la carte des fonctionnalités](#)", page 54 et "[Décrire une fonctionnalité](#)", page 56.

Pour accéder à la liste des **cartes de fonctionnalités techniques** à partir du volet de navigation **Projets en cours** :

1. Sélectionnez **Spécification des fonctionnalités et des règles** dans le menu de navigation.
2. Cliquez sur le dossier **Cartes des fonctionnalités techniques** et sélectionnez l'onglet **Cartes de fonctionnalités techniques**.
La liste des cartes de fonctionnalités techniques apparaît dans la zone d'édition.

☛ Pour accéder à la liste des fonctionnalités techniques, sélectionnez l'onglet **Fonctionnalités techniques**.

DÉCRIRE UN ENVIRONNEMENT D'APPLICATION

Dans **HOPEX Application Design**, les **applications** constituent le lien entre les fonctionnalités métier et techniques attendues et les éléments de l'architecture.






Une application est un composant logiciel déployable qui fournit un ensemble de fonctionnalités à des utilisateurs.

Dans un environnement d'application vous pouvez décrire les communications qu'une application établit avec les autres composants du projet. Chaque contexte d'utilisation d'une application est représenté par un **environnement d'application**.



Un environnement d'application présente le contexte d'utilisation des applications d'un projet. Il décrit les interactions, entre les acteurs impliqués et les applications internes et externes du projet, qui permettent d'assurer les fonctionnalités attendues du projet.

Un environnement d'application peut être décrit par différents types de diagramme :

- les **diagrammes de scénario de flux** qui décrivent les flux échangés entre les éléments applicatifs de l'environnement dans différents contextes.
 Pour plus de détails sur les diagrammes de scénario de flux, voir ["Décrire les scénarios de flux", page 154](#).
- les **diagrammes de cas d'utilisation** qui permettent de représenter un contexte d'utilisation d'un élément applicatif dans un format spécifique.
 Un scénario de flux peut aussi être associé à un cas d'utilisation. Pour plus de détails sur la description d'un cas d'utilisation, voir ["Décrire un cas d'utilisation d'un projet", page 62](#).
- les **diagrammes d'environnement d'application** qui décrivent les interactions entre les éléments applicatifs de l'environnement et l'extérieur dans le contexte représenté par l'environnement.
 Pour plus de détails sur la description d'un diagramme d'environnement d'application, voir ["Construire le diagramme d'environnement d'application", page 59](#).

Pour la description de la structure interne d'une application, voir ["Décrire une application avec HOPEX Application Design"](#).

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir ["Le volet Projet en cours", page 22](#).

Accéder à la liste des environnements d'application

Pour accéder à la liste des environnements d'application d'un projet :

1. Cliquez sur le menu de navigation puis sur **Projets en cours** :
2. Dans le volet de navigation sélectionnez **Conception fonctionnalités et règles**.

3. Cliquez sur le dossier **Liste des environnements d'application**.
La liste des environnements d'application apparaît dans la zone d'édition.

Les propriétés d'un environnement d'application


La page **Caractéristiques** des propriétés d'un environnement d'application permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de l'environnement d'application, il s'agit du projet courant,
- son **Nom**,
- le texte de sa **Description**.

Voir aussi : ["Décrire une application avec HOPEX Application Design", page 78](#).

Construire le diagramme d'environnement d'application

Avec **HOPEX Application Design**, un *environnement d'application* est totalement décrit par un diagramme d'environnement d'application qui permet de décrire les interactions entre les applications de l'environnement décrit, ses utilisateurs et les applications externes.

 *Un environnement d'application présente le contexte d'utilisation des applications d'un projet. Il décrit les interactions, entre les acteurs impliqués et les applications internes et externes du projets, qui permettent d'assurer les fonctionnalités attendues du projet.*

Le diagramme suivant décrit l'environnement d'application correspondant au traitement des achats de pièces détachées.

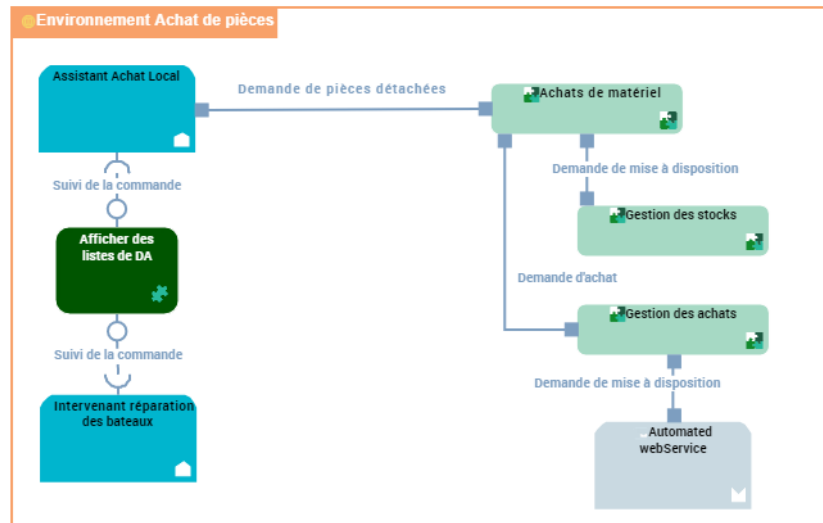


Diagramme d'environnement d'application "Achat de pièces détachées"

Les demandes d'achat de pièces détachées sont formulées par les intervenants sur les réparations de bateaux et elles sont traitées par les assistant achat locaux.

La consultation des pièces en stock est effectuée par l'assistant achat local. Suite à la consultation, l'assistant peut faire une demande de mise à disposition.

Deux types de commandes sont possibles; une commande de pièces déjà référencées ou une commande de pièces non référencées. Dans les deux cas, une demande de mise à disposition est effectuée.

Le suivi de la commande est assuré à la fois par l'assistant achat local et l'intervenant chargé de la réparation des bateaux.

Un diagramme d' environnement d'application est composé :

- d'**applications** qui représentent l'environnement décrit.

Dans l'exemple, il s'agit des applications utilisées pour l'achat des pièces détachées.



Une application est un composant logiciel déployable qui fournit un ensemble de fonctionnalités à des utilisateurs.

- d'**applications**, de **services applicatifs** ou de **micro-services partenaires** qui représentent les éléments externes utilisés dans l'environnement décrit.

Dans l'exemple, il s'agit des services web automatisés.



Un service applicatif est l'élément de découpage d'une application qui est mis à la disposition de l'utilisateur final de cette application dans le cadre de son travail.

- d'**acteurs** ou de **positions type** qui représentent les utilisateurs ou les fournisseurs de l'environnement décrit.

Dans l'exemple, il s'agit des intervenants locaux.



Un acteur représente le rôle joué par quelque chose ou quelqu'un se trouvant dans l'environnement de l'entreprise ou du système étudié. Il est en relation avec le métier de l'entreprise et interagit avec le système dans différents cas d'utilisation. Ce peut être un élément de la structure de l'entreprise tel qu'une direction, un service ou un poste de travail.

- **interactions** entre les composants.



Une interaction représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par un contrat d'échange.

- points d'accès, de requête et de service

Pour plus de détails sur la description de ces éléments, voir ["Gérer les interactions"](#).


DÉCRIRE UN CAS D'UTILISATION D'UN PROJET

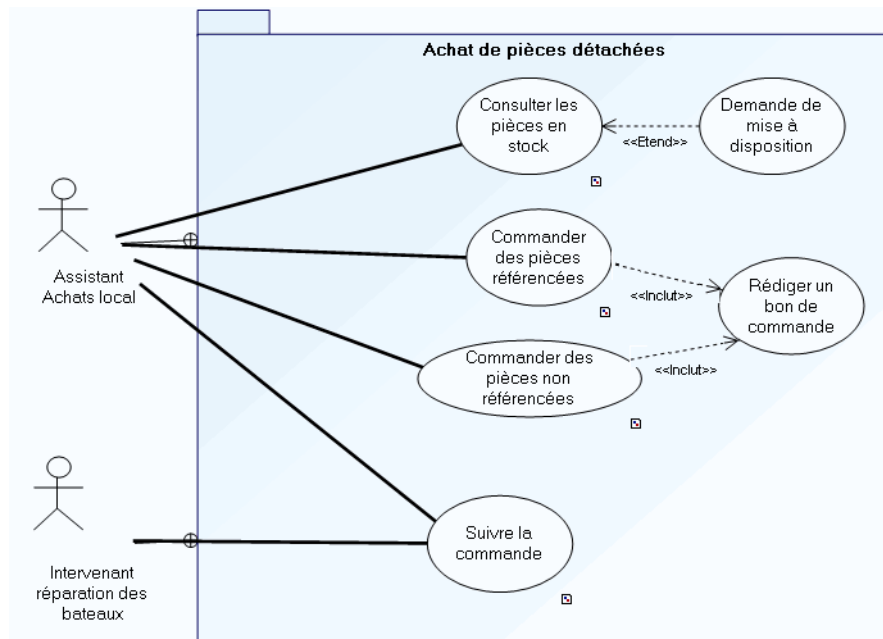
Créer les cas d'utilisation des composants du projet permet de représenter les différents contextes d'utilisation du système. A partir d'un cas d'utilisation, vous pouvez décrire comment une fonctionnalité, ou un ensemble de fonctionnalités, est implémentée au sein d'une application ou d'un service.

Ainsi, au travers de cas d'utilisation, **HOPEX Application Design** permet de compléter la description d'une architecture applicative par des spécifications détaillées de ses différents sous-systèmes.

Exemple de cas d'utilisation

Un diagramme de *cas d'utilisation* permet de décrire les interactions entre un système et les acteurs de l'organisation.

 Un cas d'utilisation est une suite d'actions qui amène un résultat observable pour un acteur particulier. Des scénarios illustrent les cas d'utilisation par l'exemple.



Le système est utilisé pour la consultation des pièces en stock et la commande de nouvelles pièces détachées.

La consultation des pièces en stock est effectuée par l'assistant achat local. Suite à la consultation, l'assistant peut faire une demande de mise à disposition.

Deux types de commandes sont possibles; une commande de pièces déjà référencées ou une commande de pièces non référencées. Dans les deux cas, un bon de commande doit être rédigé.

Le suivi de la commande est assuré à la fois par l'assistant achat local et l'intervenant chargé de la réparation des bateaux.

Créer un diagramme de cas d'utilisation sur un environnement d'application

Condition préalable

- Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.
- Avoir créé un environnement d'application. Voir "[Décrire un environnement d'application](#)", page 58.

Pour créer un diagramme de cas d'utilisation à partir d'un environnement d'application :

1. Cliquez sur le menu de navigation puis sur **Projets en cours**.
2. Dans le volet de navigation cliquez sur **Conception fonctionnalités et règles**.
3. Cliquez sur la tuile **Environnements d'application**.
La liste des environnements d'application apparaît dans la zone d'édition
4. Faites un clic droit sur l'environnement d'application concerné et sélectionnez **Nouveau > Diagramme de cas d'utilisation de l'environnement d'application**.
Le diagramme créé s'ouvre dans la fenêtre d'édition.

Les éléments du diagramme de cas d'utilisation

Pour la description complète du diagramme de cas d'utilisation, voir "[Le diagramme de cas d'utilisation](#)".

GÉNÉRER LE RAPPORT D'ANALYSE FONCTIONNELLE


Dans **HOPEX Application Design** un type de rapport MS Word décrit la spécification fonctionnelle de projets IT. Le rapport s'appuie notamment sur la modélisation des cas d'utilisation, des acteurs et des processus applicatifs du projet.

Pour être visibles dans le rapport, les éléments modélisés doivent faire partie du périmètre de l'analyse fonctionnelle ; ils doivent donc être rattachés aux paquetages ayant pour stéréotype "groupe de cas d'utilisation", "groupe des modèles de données", "groupe des interfaces utilisateur" ou "groupe des profils d'acteurs".

Lorsque les informations existent, le rapport généré présente :

- les acteurs du système (acteurs système ou utilisateurs). Il s'agit des acteurs contenus dans le paquetage "*Nom du projet* - Acteurs" ou tout autre paquetage de stéréotype "Groupe de profils d'acteurs" rattaché au projet.
- les diagrammes de cas d'utilisation complétés par la description de chacun des cas d'utilisation. Il s'agit des cas d'utilisation contenus dans le paquetage "*Nom du projet* - Cas d'utilisation" ou tout autre paquetage de stéréotype "Groupe de cas d'utilisation" rattaché au projet.
- la dynamique des interfaces utilisateurs représentée par les diagrammes de processus applicatifs contenus dans le paquetage "*Nom du projet* - IHM" ou tout autre paquetage de stéréotype "Groupe des interface utilisateur" rattaché au projet.
- les modèles des données métier contenus dans un paquetage de stéréotype "Modèle de données", sous le paquetage des cas d'utilisation du projet.
- les diagrammes de machine à états contenus dans un paquetage de stéréotype "Modèle de données", sous le paquetage des modèles de données du projet.

Pour créer un rapport d'analyse fonctionnelle d'un projet de spécification (AD) :

1. Cliquez sur le menu de navigation puis sur **Projets en cours**.
 L'accès au volet de navigation **Projet en cours** nécessite que vous ayez défini un environnement de travail sur le projet concerné. Voir "[Le volet Projet en cours](#)", page 22.
2. Dans le volet de navigation cliquez sur **Conception fonctionnalités et règles**.
3. Dans la zone d'édition, cliquez sur la tuile **Rapport d'analyse fonctionnelle**.
4. Cliquez sur **Nouveau**.
L'assistant de création du rapport apparaît.
5. Indiquez le nom du rapport.
6. Cliquez sur **Suivant** puis sur **OK**.

Pour ouvrir le document généré :

- 1 Cliquez sur l'icône du document et sélectionnez **Documentation** > **Ouvrir**.



DÉCRIRE LES DONNÉES D'UN PROJET



Dans **HOPEX Application Design**, la description de l'architecture technique d'un système applicatif peut être complétée par celle des données utilisées par le système.

Les points suivants présentent comment modéliser les données utilisées par les processus et les applications, sans tenir compte de leur implémentation physique.

- ✓ ["Modéliser les données du projet", page 68,](#)
- ✓ ["Utiliser les dépôts de données", page 72.](#)

MODÉLISER LES DONNÉES DU PROJET

HOPEX Application Design fournit les outils nécessaires à la modélisation des données logiques, via les diagrammes de classes et les modèles de données.

A travers les concepts de *Domaine de données* et de *Vue de données*, vous pouvez détailler une structure de données logique dans un contexte d'utilisation particulier.

➡ Pour plus de détails sur la création et la mise à jour d'un modèle de données, voir dans le guide **HOPEX Information Architecture** le chapitre "Le modèle données".

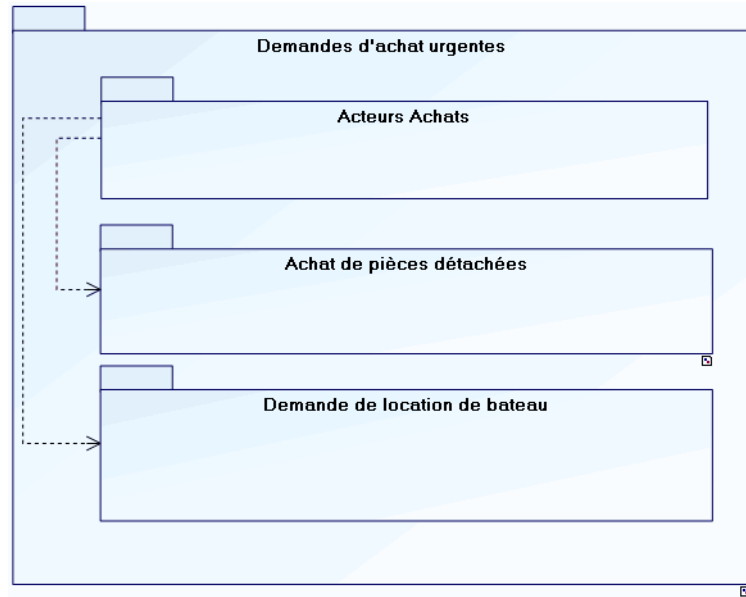
Les différents concepts de la vue logique

Paquetage (UML)

Un paquetage permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et les relations qui existent entre eux.

Le paquetage est un élément détenteur, il fournit un espace de nommage pour les éléments qu'il regroupe.

Le paquetage vous permet de classer les éléments référencés dans un projet. Vous pouvez créer des sous-paquetages dans un paquetage afin de classer plus finement les objets, par exemple les acteurs d'un projet.



Les demandes d'achat urgentes sont prévues pour traiter les achats de pièces détachées et les demandes de location de bateau. Dans ces deux cas, les utilisateurs sont des acteurs du domaine des achats.

La représentation graphique des éléments d'un paquetage se fait dans un diagramme de classes.

Plus plus de détails sur la construction d'un diagramme de classes, voir "[Le diagramme de classes](#)".

Voir aussi "[Les diagrammes de structure et de déploiement](#)", page 73.

Modèles de données

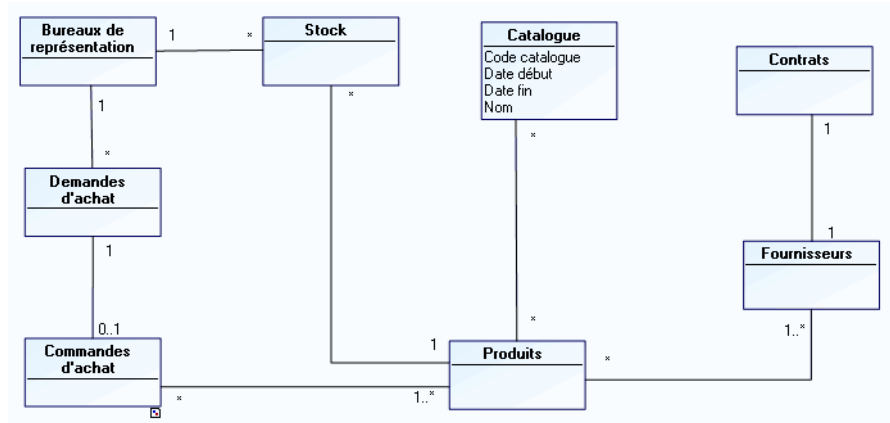
Tout comme le paquetage, le modèle de données permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et les relations qui existent entre eux.

La représentation graphique des éléments d'un modèle de données se fait dans un diagramme de données.

Pour plus de détails sur la création et la mise à jour d'un modèle de données voir "[Le modèle de données](#)", page 9.

Exemple

Le modèle de données du projet "Automatisation des demandes d'achat" est présenté ci-dessous.



L'application gère les demandes d'achat, les commandes et les stocks de produits dans chacun des bureaux de représentation.

Un catalogue centralisé des produits et des fournisseurs est mis en place.

Les contrats avec les fournisseurs référencés sont également accessibles depuis l'application.

➡ Pour plus de détails sur la création et la mise à jour d'un modèle de données, voir "Modéliser les données".

Les domaines de données

Un domaine de données représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles (voir "[Utiliser les dépôts de données](#)", page 72). Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.

Un domaine de données logique permet de définir une structure de données logiques constituée de classes et de vue de classes.

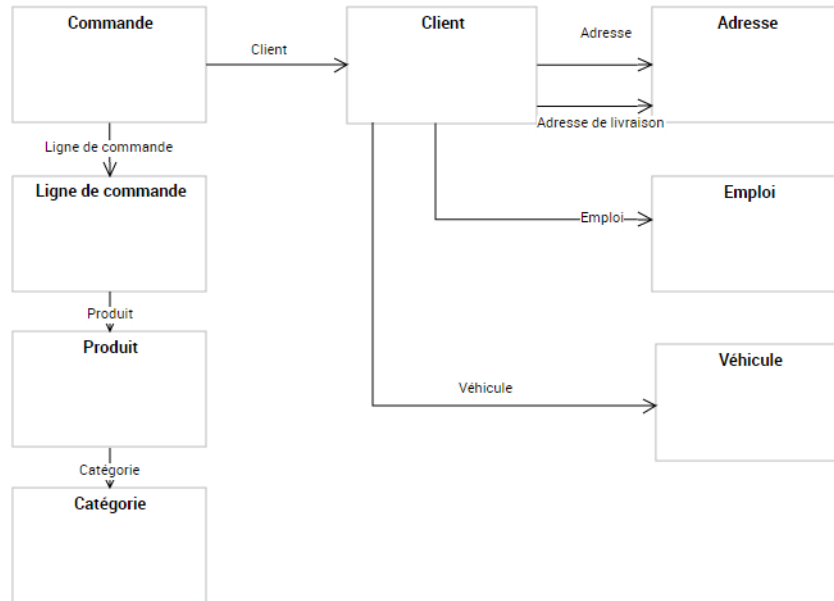
Un domaine de données logique est détenu par un paquetage, il peut référencer des objets détenus dans d'autres paquetages.

Il est possible de définir le mode d'accès (création, suppression, etc.) aux objets référencés par un domaine de données en les intégrant comme composants du domaine de données.

Lors d'une intégration avec HOPEX Database Builder, une structure physique correspondante peut être définie via un domaine de données physique. Celui-ci est constitué de tables et de vues de tables.

Exemple

Le diagramme de domaine de données suivant représente une structure de données relative aux Commandes ; il décrit sous forme de tout/partie les classes et leurs relations.



Pour répondre à des cas d'utilisation précis, vous pouvez créer des *Vues de données* dans lesquelles vous pouvez visualiser et modifier le périmètre couvert par des classes.

Vue de données logique

Une vue de données permet de représenter le périmètre couvert par un élément de modèle de données ou de domaine de données. Une vue de données est construite à partir d'une sélection de plusieurs classes reliées dans le contexte spécifique de la vue.

Dépôt de données

Un dépôt de données permet de référencer dans un processus ou un système applicatif des données persistantes, définies dans un domaine de données.

Voir ["Utiliser les dépôts de données"](#), page 72.

UTILISER LES DÉPÔTS DE DONNÉES



Un dépôt de données fournit un mécanisme pour mettre à jour ou consulter des données qui vont persister au delà de la portée du processus en cours. Il permet de stocker les données des flux en entrée et de les retransmettre par l'intermédiaire d'un ou plusieurs flux en sortie.

Introduction au concept de dépôt de données

Un dépôt de données référence un **domaine de données**.



Un domaine de données représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles. Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.



*Pour plus de détails sur les domaines de données, voir le chapitre "Les domaines de données logiques et applicatifs" du guide **HOPEX Information Architecture**.*

Si vous décrivez un système applicatif logique, seuls des **dépôts de données logiques** peuvent être utilisés.



Un dépôt de données logique matérialise l'utilisation des données par les systèmes applicatifs sans considérer comment leur accès sera concrètement mis en œuvre.

Si vous décrivez un système applicatif, seuls des **dépôts de données physiques** peuvent être utilisés.



Un dépôt de données physique représente l'implémentation d'un dépôt de données logique.

Si vous décrivez un de scénario de séquences ou de scénario de flux, vous ne pouvez utiliser que les **dépôts de données applicatives**.



Un dépôt de données applicatives matérialise l'utilisation de données dans le contexte d'un composant logiciel (par exemple une application). Un dépôt de données applicatives fournit un mécanisme pour mettre à jour ou consulter des données stockées à l'intérieur du composant logiciel en cours.



Les diagrammes de scénario de flux qui décrivent les flux échangés dans différents scénarios d'utilisation de l'objet décrit.



Les diagrammes de scénario de séquences de flux qui décrivent la chronologie des flux échangés dans différents scénarios d'utilisation de l'objet décrit.

Enfin, vous pouvez également distinguer les dépôts de données locaux à un système des dépôts de données externes qui sont positionnés à la frontière des diagrammes.



Un dépôt de données local représente un dépôt de données utilisé uniquement à l'intérieur du système décrit.




Un dépôt de données externe représente un dépôt de données utilisé à l'intérieur et à l'extérieur du système décrit.

Les contextes d'utilisation

Le tableau ci-dessous présente la liste des diagrammes qui utilisent les différents types de dépôts de données.


Type de dépôt de données	Diagrammes
Dépôt de données logique	Diagrammes de structure de système applicatif logique
Dépôt de données physique	Diagrammes de structure <ul style="list-style-type: none"> - d'application, - de système applicatif, - de services applicatifs, - de micro-services.
Dépôt de données applicatives	Diagrammes de scénario de séquence <ul style="list-style-type: none"> - d'application, - de système applicatif, - de services applicatifs, - de micro-services, Diagrammes de scénario de flux <ul style="list-style-type: none"> - d'application, - de système applicatif, - de service applicatif, - de micro-service.


Créer un dépôt de données local

 *Un dépôt de données local représente un dépôt de données utilisé uniquement à l'intérieur du système décrit.*

Pour créer, par exemple, un **dépôt de données physique local** à partir d'un diagramme de structure de système applicatif :

1. Ouvrez le diagramme qui vous intéresse.
2. Dans la barre d'objets du diagramme, cliquez sur le bouton **Dépôt de données physique local**.
3. Cliquez à l'intérieur du cadre du système applicatif décrit.
Une fenêtre de création vous demande de choisir un **Domaine de données** existant.

 *Le domaine de données représente la structure qui va supporter concrètement le dépôt de données. Un dépôt de données physique ou applicatif est supporté par un **domaine de donnée applicatif**. Un dépôt de données logique est supporté par un **domaine de données logiques**.*

 *Pour plus de détails sur les domaines de données, voir le chapitre "Les domaines de données logiques et applicatifs" du guide **HOPEX Information Architecture**.*

4. Sélectionnez un **Domaine de données** existant.
5. Cliquez sur **OK**.
Le dépôt de données physique local apparaît dans le diagramme, il porte le nom du domaine de données associé.

Créer un dépôt de données externe



Un dépôt de données externe représente un dépôt de données utilisé à l'intérieur et à l'extérieur du système décrit.

Pour créer, par exemple, un dépôt de données physique externe à partir d'un diagramme de structure de système applicatif :

1. Ouvrez le diagramme qui vous intéresse.
2. Dans la barre d'objets du diagramme, cliquez sur le bouton **Dépôt de données physique externe**.
3. Cliquez sur le bord du cadre du système applicatif décrit.
Une fenêtre de création vous demande de choisir un **Domaine de données** existant.

☛ *Le domaine de données représente la structure qui va supporter concrètement le dépôt de données. Un dépôt de données physique ou applicatif est supporté par un **domaine de donnée applicatif**. Un dépôt de données logique est supporté par un **domaine de données logiques**.*

☛ *Pour plus de détails sur les domaines de données, voir le chapitre "Les domaines de données logiques et applicatifs" du guide **HOPEX Information Architecture**.*

4. Sélectionnez un **Domaine de données** existant.
5. Cliquez sur **OK**.

Le dépôt de données physique local apparaît dans le diagramme, il porte le nom du domaine de données associé.

Décrire l'accès à un dépôt de données

Pour créer un accès en lecture au dépôt de données :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Lien**
2. Tirez un lien entre le dépôt de données et l'entité qui lit les données (composant ou utilisation de système applicatif).
Un **Accès en lecture au stockage de données** est automatiquement créé avec le lien orienté du dépôt de données vers l'entité est créé.

☛ *Pour créer un lien en écriture, vous devez tirer un lien entre l'entité qui écrit et le dépôt de données. Un **Accès en écriture au stockage de données** est alors automatiquement créé.*

DÉCRIRE LES SPÉCIFICATIONS TECHNIQUES D'UN PROJET



Après les spécifications fonctionnelles et la description des données utilisées, les spécifications techniques du projet visent à décrire l'architecture technique des composants du projet : applications, services, bases de données et entités touchées par le projet.

DÉCRIRE UNE APPLICATION AVEC HOPEX APPLICATION DESIGN

Une application est un ensemble de composants logiciels qui constituent un tout cohérent au regard de leur déploiement, de leur couverture fonctionnelle et des techniques informatiques utilisées.

L'application est l'unité de gestion et de déploiement d'un ensemble de composants logiciels. Elle est susceptible d'être déployée sur une ou plusieurs machines.

Un projet de description de l'architecture fonctionnelle d'un système d'information permet de recenser les applications existantes et leurs échanges. Voir "[Décrire les spécifications fonctionnelles d'un projet](#)", page 53.



Une application est un composant logiciel déployable qui fournit un ensemble de fonctionnalités à des utilisateurs.

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.


Créer une application avec HOPEX Application Design

Pour créer une application :

1. Cliquez sur le menu de navigation puis sur **Projet en cours**.
2. Dans le volet de navigation cliquez sur **Conception de l'architecture applicative**.
3. Cliquez sur la tuile **Applications**.
La liste des applications apparaît dans la zone d'édition.
4. Cliquez sur **Nouveau**.
La fenêtre **Création d'une application** apparaît.
5. Saisissez le **Nom** de votre application et cliquez sur **OK**.
La nouvelle application apparaît dans la liste.

Propriétés d'une application avec HOPEX Application Design





Pour accéder aux propriétés d'une application :

1. Dans la liste des applications, sélectionnez l'application voulue et cliquez sur le bouton **Propriétés**  de la fenêtre d'édition.

Différentes pages vous donnent accès aux caractéristiques et aux composants de l'application.

Modéliser une application avec HOPEX Application Design

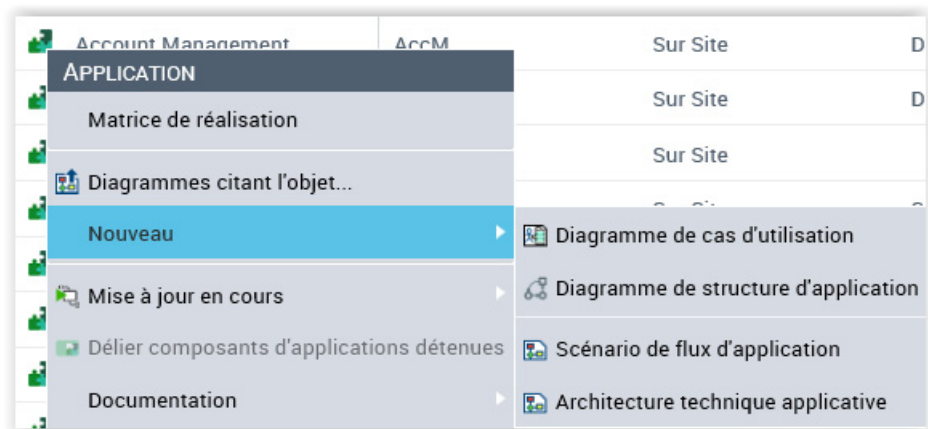
Avec **HOPEX Application Design** une application peut-être décrite par plusieurs types de diagramme :

- un *diagramme de structure d'application* : permet de représenter les interactions entre les différents composants de l'application sous forme de contrats d'échanges.
 Pour plus de détails, voir "[Créer le diagramme de structure d'application](#)", page 152.
- un *diagramme d'architecture technique d'application*.
Une architecture technique d'application décrit une des configurations possibles de déploiement d'une application. Elle décrit comment les différentes zones techniques de l'application sont reliées entre elles et les technologies et les protocoles de communication qu'elles utilisent. Une application peut avoir plusieurs architecture techniques possibles (Ex: installation autonome, déploiement horizontal ou vertical, etc.)
 Vous pouvez accéder à la liste des architectures techniques applicatives du projet courant sous le volet de navigation **Conception de l'architecture applicative**.
 Pour plus de détails sur l'architecture technique d'application, voir "[Créer un diagramme d'architecture technique d'application](#)", page 160.
- un *scénario de flux d'application* : présente les flux échangés entre les services applicatifs ou les micro-services utilisés par cette application. Un scénario peut représenter un cas d'utilisation particulier de l'application ou plus globalement l'ensemble des flux échangés à l'intérieur de cette application.
 Pour plus de détails, voir "[Utiliser un diagramme de scénario de flux d'application](#)", page 154.


Pour créer le diagramme d'une application :


1. Affichez la liste des applications.

2. Cliquez sur l'icône de l'application voulue et sélectionnez **Nouveau** suivi du type de diagramme à créer.



DÉCRIRE LES SERVICES APPLICATIFS ET LES MICRO-SERVICES

 Un service applicatif est l'élément de découpage d'une application qui est mis à la disposition de l'utilisateur final de cette application dans le cadre de son travail.

 Un micro-service est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Définir un service applicatif ou un micro-service

Accéder à la liste des services applicatifs

Pour accéder à la liste des services applicatifs à partir du volet de navigation **Projet en cours** :

- 1 Sélectionnez **Conception de l'architecture applicative > Services applicatifs**.

La liste des services applicatifs apparaît dans la zone d'édition.

Accéder à la liste des micro-services

Pour accéder à la liste des micro-services à partir du volet de navigation **Projet en cours** :

- 1 Sélectionnez **Conception de l'architecture applicative > Micro-services**.

La liste des micro-services apparaît dans la zone d'édition.

Propriétés d'un service applicatif ou d'un micro-service

La description complète d'un service applicatif, ou d'un micro-service, est accessible à partir de ses pages de propriétés.

La page **Caractéristiques** des propriétés d'un micro-service permet d'accéder à :

- son **Détenteur**, par défaut lors de la création du micro-service, il s'agit de la bibliothèque courante.
- son **Nom**,
- le texte de sa **Description**,
- la section **Technologies** permet d'accéder à la liste des *technologies logicielles* utilisées par les micro-service.

☛ Pour plus de détails sur les technologies logicielles, voir le guide **HOPEX IT Architecture**.

Utiliser un scénario de flux de service applicatif ou de micro-service

☛ Pour plus de détails sur les diagrammes de scénario de flux, voir *"Décrire les scénarios de flux", page 154*.

Un scénario de flux de micro-service permet de représenter les flux échangés entre certains éléments du micro-service dans un contexte donné. Les éléments représentés sont :

- des services applicatifs,
- des micro services,
- des dépôts de données applicatives locaux ou externes,
- des ports applicatifs en entrée ou en sortie.

Les échanges proposés entre ces éléments :

- des flux applicatifs qui portent un contenu,
- des canaux de flux applicatifs qui regroupent plusieurs flux applicatifs sur un même lien,
- des canaux de données applicatives qui représentent les échanges effectués avec les dépôts de données applicatives.

Utiliser le diagramme de structure de service applicatif ou de micro-service

☛ Pour plus de détails sur les diagrammes de structure, voir *"Créer un diagramme de structure", page 152*.

Avec **HOPEX Application Design**, les composants d'un micro-service peuvent être décrits par un diagramme de structure de micro-service.

Ce type de diagramme est composé de :

- des services applicatifs,
- des micros services,
- des dépôts de données physiques externes ER ou BDR, voir "[Utiliser les dépôts de données](#)", page 72.
- points d'accès, de requête et de service, voir "[Décrire les points de service et de requête](#)", page 165.
- *interactions* entre les composants.



Une interaction représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par un contrat d'échange.

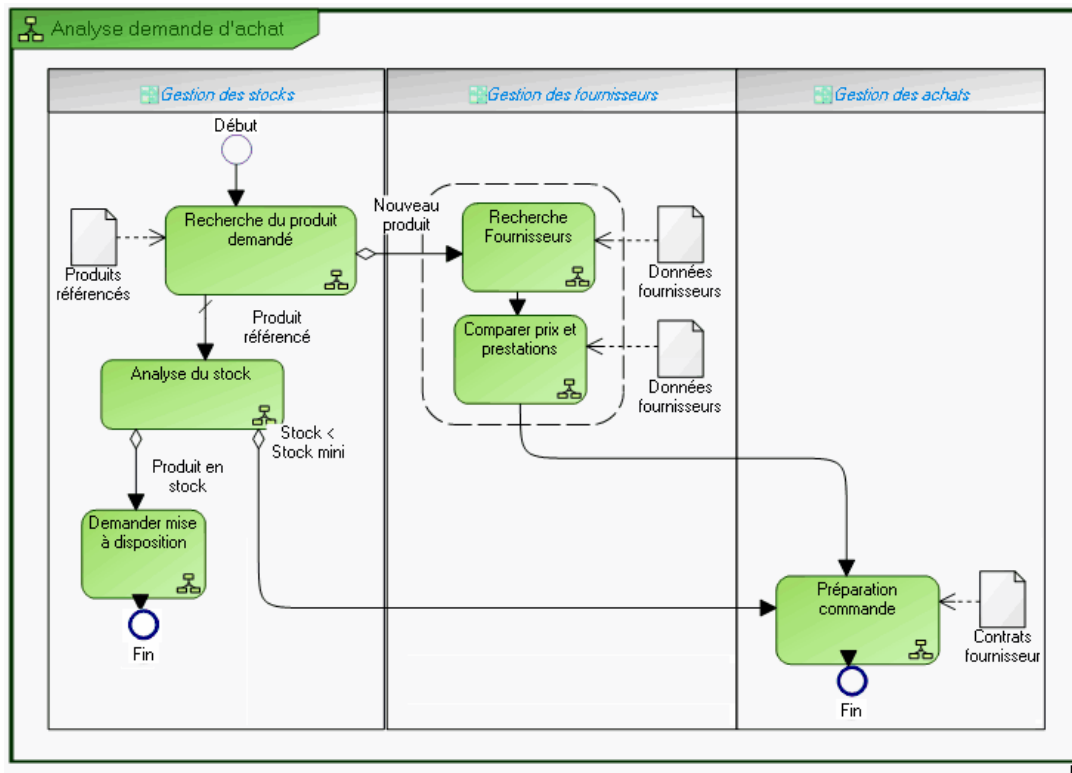
DÉCRIRE LES PROCESSUS APPLICATIFS

En phase de spécification détaillée, le déroulement des tâches mises en oeuvre dans un service applicatif peut également être modélisé par un processus applicatif. Plus généralement, le fonctionnement d'un élément d'architecture peut être décrit par un processus applicatif qui modélise, par exemple, l'enchaînement des écrans présentés à l'utilisateur.

Exemple de processus applicatif

Le diagramme ci-après représente le traitement des demandes d'achat.

- Une recherche du produit est effectuée à partir de la base des produits Référencés.
- Si le produit est nouveau, une recherche de fournisseur et une étude comparative des prix est menée. Puis une commande est émise et le processus prend fin.
- Si le produit est référencé, l'étude de l'état des stocks est réalisée.
- Si le stock est suffisant, une demande de mise à disposition est activée et le processus prend fin.
- Si le stock est inférieur au stock minimum, une commande est émise vers le fournisseur et le processus prend fin.




Processus applicatif de "Demande d'achat"

➡ Pour plus de détails sur l'utilisation des processus applicatifs voir le guide HOPEX Business Process Analysis, chapitre "Les processus applicatifs", page 83.

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Gérer les processus applicatifs avec HOPEX Application Design

 Un processus applicatif est la représentation exécutable d'un processus. Les éléments formalisant un processus applicatif sont les suivants : les événements du workflow, les tâches à accomplir durant le traitement, les éléments algorithmiques d'enchaînement des tâches, les flux d'information échangés avec les participants.

➡ Pour plus de détails sur la construction d'un diagrammes de processus applicatif, voir dans le guide **HOPEX Business Process Analysis** le paragraphe "Gérer un processus applicatif".



Accéder aux processus applicatifs

Pour accéder à la liste des processus applicatifs à partir du volet de navigation **Projet en cours** :

1. Sélectionnez **Conception de l'architecture applicative**.
2. Sélectionnez la tuile **Processus applicatifs**.
La liste des processus applicatifs apparaît.

Créer un diagramme de processus applicatif

Pour créer un diagramme de processus applicatif :

1. Faites un clic droit sur le nom du processus applicatif qui vous intéresse et sélectionnez **Nouveau > Diagramme de processus applicatif**.
Le diagramme s'ouvre dans la zone d'édition.

➡ L'initialisation du diagramme positionne automatiquement le cadre du processus décrit et les événements principaux.

Modéliser les tâches d'un processus applicatif

La phase d'analyse fonctionnelle du projet consiste à s'appuyer sur les cas d'utilisation ainsi que sur le modèle de données, obtenus en phase d'étude préalable, pour représenter l'architecture fonctionnelle du futur système.

La phase d'analyse fonctionnelle décrit les processus applicatifs mis en oeuvre dans les différents cas d'utilisation d'une application ou d'un service.

Un diagramme de processus applicatif va spécifier l'enchaînement des tâches à réaliser de manière à ce que l'utilisateur puisse vérifier que l'application décrite réponde à son besoin.

Décrire un cas d'utilisation par un processus applicatif

Pour créer un processus applicatif à partir d'un cas d'utilisation :

1. Faites un clic droit sur le cas d'utilisation pour faire apparaître son menu contextuel.
2. Sélectionnez **Nouveau > Processus applicatif**
Le diagramme d'un nouveau processus applicatif s'affiche. Le processus associé au cas d'utilisation est positionné dans le diagramme.

☛ Pour plus de détails sur la construction d'un diagramme de processus applicatif, voir dans le guide **HOPEX Business Process Analysis** le paragraphe "Gérer un processus applicatif".

Exemple de modélisation fonctionnelle

Les *processus applicatifs* utilisés pour l'analyse fonctionnelle d'un projet sont rangés dans un paquetage.

Dans l'exemple du projet d'automatisation du traitement des demandes d'achat les processus applicatifs sont rangés dans le paquetage "Demandes d'achat urgentes".

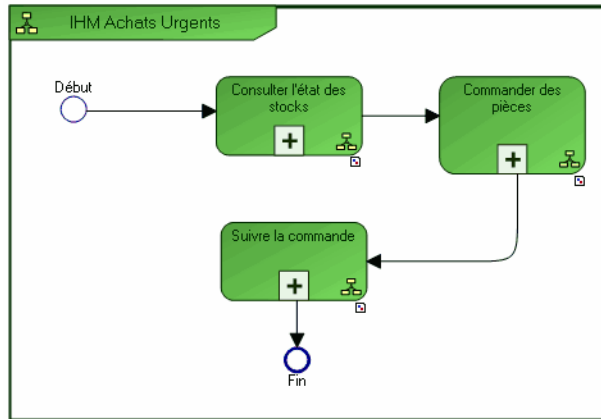
📖 Un processus applicatif est la représentation exécutable d'un processus. Les éléments formalisant un processus applicatif sont les suivants : les événements du workflow, les tâches à accomplir durant le traitement, les éléments algorithmiques d'enchaînement des tâches, les flux d'information échangés avec les participants.

Accéder à un diagramme de processus applicatif à partir d'un paquetage

Pour accéder à un diagramme de processus applicatif à partir d'un paquetage :

1. Dans la fenêtre de navigation **Accueil**, déployez l'arbre associé à ce paquetage, par exemple le paquetage "Demande d'achat urgentes". La liste des processus applicatifs rangés dans le paquetage s'affiche.

2. Faites un clic droit sur le processus applicatif, par exemple "IHM Achats Urgents" pour faire apparaître son menu contextuel et sélectionnez **Diagramme de processus applicatif**.
Le diagramme s'ouvre.



Les principales étapes de ce processus applicatif sont :

- consulter l'état des stocks locaux afin qu'une pièce détachée puisse être mise à disposition
- commander une pièce dans le cas où le stock aurait atteint un seuil critique ou si la pièce n'est pas référencée
- suivre une commande passée.

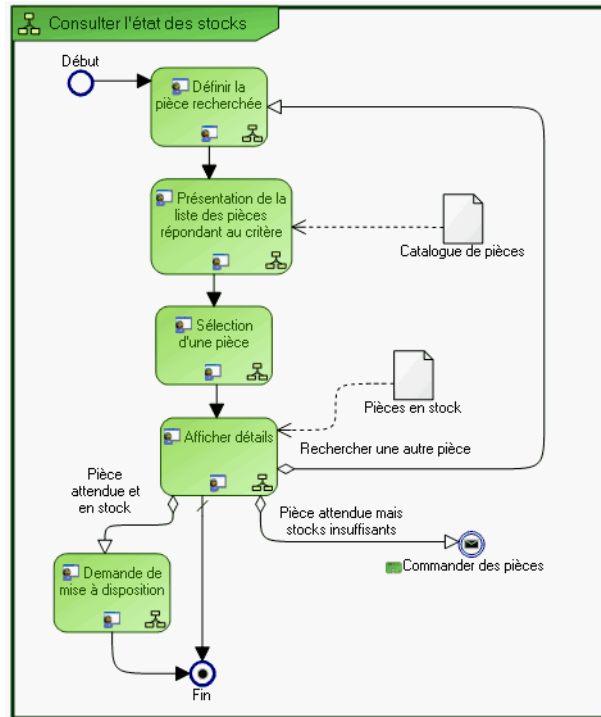
☛ Vous pouvez également accéder à un processus applicatif depuis un cas d'utilisation. Pour plus de détails, voir "[Décrire un cas d'utilisation par un processus applicatif](#)", page 87.

Afficher le diagramme d'une étape de processus applicatif

Pour afficher le diagramme décrivant en détail une étape du processus applicatif :

1. Cliquez sur le processus applicatif, par exemple "Consulter l'état des stocks" pour faire apparaître son menu contextuel.

2. Sélectionnez **Diagramme de processus applicatif**.
Le diagramme associé au processus s'ouvre.



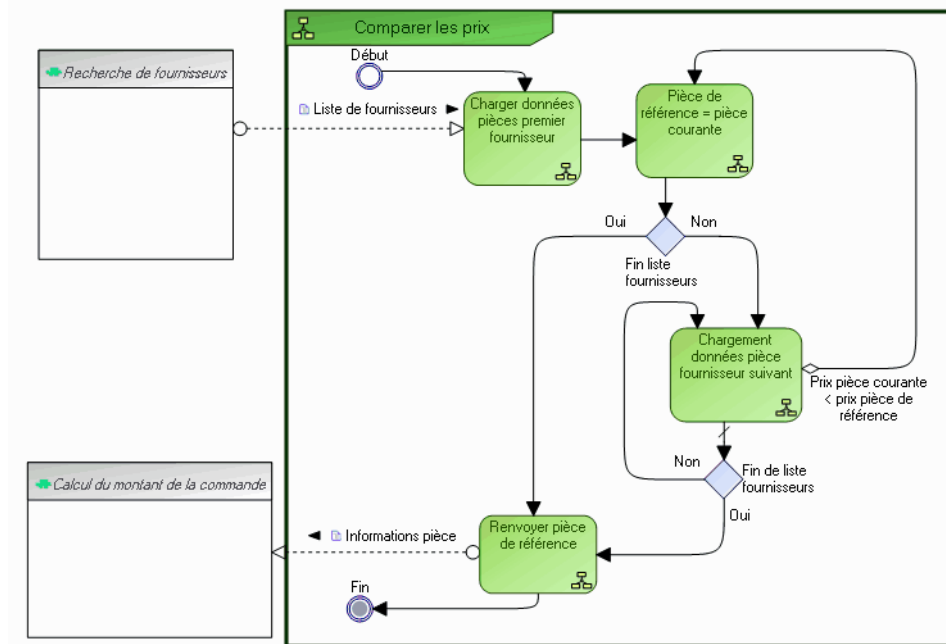
La consultation de l'état des stocks commence par l'affichage d'un écran permettant l'identification de la pièce recherchée. La liste des pièces trouvées dans le catalogue est présentée dans l'écran suivant.

Quand l'utilisateur a sélectionné la pièce qui l'intéresse, les informations de détails sont affichées. A partir de cet écran, il est possible de rechercher des informations sur une autre pièce, ou bien de demander la mise à disposition de la pièce, ou bien encore de commander cette pièce.

Modéliser les tâches d'un service applicatif

La phase d'analyse détaillée des composants système impactés par projet consiste à modéliser finement le fonctionnement des services applicatifs.

Dans le contexte de l'exemple d'automatisation du traitement des demandes d'achat urgentes, le service de comparaison des prix est représenté par un processus applicatif.



Ce diagramme décrit l'algorithme du service "Comparer les prix" qui doit retourner la référence de la pièce dont le prix est le plus bas.

La liste des fournisseurs de la pièce recherchée est donnée en entrée. La pièce proposée par le premier fournisseur de cette liste devient la pièce de référence. Ensuite, et tant que la liste des fournisseurs n'est pas vide, les données concernant la pièce recherchée sont analysées. Si le prix de la pièce courante est moins élevé que le prix de la pièce de référence, la pièce de référence devient la pièce courante. Quand la liste complète des fournisseurs a été analysée, les informations concernant la pièce de référence sont envoyées vers le service "Calcul du montant de la commande".

DÉCRIRE LES TRAITEMENTS DE LOT


HOPEX Application Design vous permet de décrire l'enchaînement des traitements automatisés dans des **diagrammes de structure de chaîne de traitement de lot**.


Ce type de diagramme permet de représenter le planning d'exécution des batch, les programmes de batch et leur organisation.

Définition d'un traitement de lot

Un traitement de lot est un ensemble de traitements informatiques exécutés par un ordinateur sans intervention humaine, généralement la nuit ou le week-end.

Un **Traitement de lot** est décrit par un **Planning d'exploitation** ou par un **Programme**.

 Un *planning d'exploitation* définit l'ensemble des traitements informatiques à effectuer sur une ou plusieurs machines pendant une période de temps donnée.

 Un *programme* est une étape élémentaire d'exécution d'un traitement de lot qui consiste à lancer l'exécution d'un programme avec les paramètres appropriés.


Un **Planning d'exploitation** est un ensemble de **Traitements de batch**. Chacun d'eux est associé soit à un **Programme** soit à un autre **Planning d'exploitation**. Un **Planning d'exploitation** est un décrit par un **diagramme de structure de chaîne de traitement de lot**.

➡ Pour plus de détails voir "[Construire un diagramme de structure de chaîne de traitement de lot](#)", page 91.

Un **Programme** est un ensemble de **Traitements de batch**. Chacun d'eux ne peut être associé qu'à un **Programme**. Un **Programme** est un décrit par un **diagramme de structure de programme batch**.

➡ Pour plus de détails voir "[Créer un diagramme de structure de programme batch](#)", page 93.

Construire un diagramme de structure de chaîne de traitement de lot

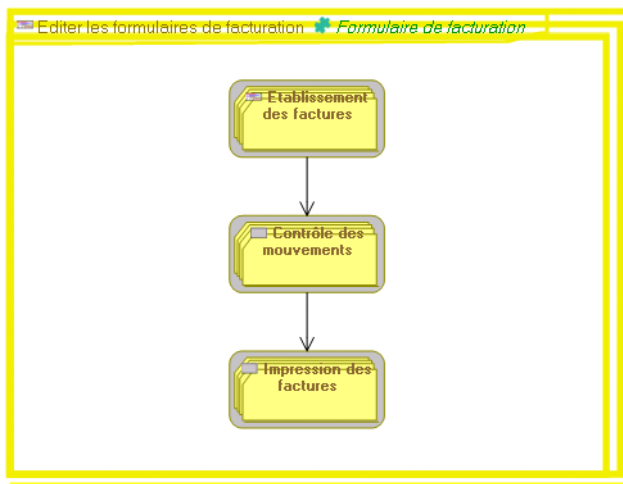
 Un *planning d'exploitation* définit l'ensemble des traitements informatiques à effectuer sur une ou plusieurs machines pendant une période de temps donnée.

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Créer un diagramme de structure de chaîne de traitement de lot

L'enchaînement de traitements automatisés peut être décrit dans un **diagramme de structure de chaîne de traitement de lot**.



Pour créer un diagramme de structure de chaîne de traitement de lot :

1. Cliquez sur le volet de navigation **Projet en cours**.
2. Dans le volet de navigation cliquez sur **Conception de l'architecture applicative**.
3. Cliquez sur la tuile **Lots**.
La liste des plannings d'exploitation apparaît.
4. Cliquez avec le bouton droit sur le planning d'exploitation qui vous intéresse et sélectionnez **Nouveau > Diagramme de structure de chaîne de traitement de lot**.
Le diagramme s'ouvre.

Ajouter un appel de traitement de batch dans le diagramme

Les composants d'un **Planning d'exploitation** sont définis à partir des **Appels de traitement de batch** qui sont positionnés dans le diagramme. Il peut s'agir de planning d'exploitation ou de programmes.

Pour ajouter un composant de type Planning d'exploitation dans le diagramme de structure de chaîne de traitement de lot :

1. Sélectionnez le bouton **Appel de traitement de batch** et cliquez dans le diagramme.
La fenêtre **d'ajout d'un appel de traitement de batch** s'ouvre.
2. Cliquez sur la flèche à droite du champ **Traitement de lot appelé** et sélectionnez **Relier** dans la liste déroulante.
Une fenêtre de connexion apparaît.
3. Sélectionnez **Planning d'exploitation** dans la partie gauche de la fenêtre et cliquez sur la flèche de recherche.
La liste des planning d'exploitation apparaît.


4. Sélectionnez le planning qui vous intéresse et cliquez sur le bouton **Relier**.
5. Cliquez sur **OK**.
L'appel de traitement batch apparaît dans le diagramme avec l'icône d'un planning d'exploitation.

Définir les séquences des batches

Pour préciser l'ordre d'exécution des traitements :

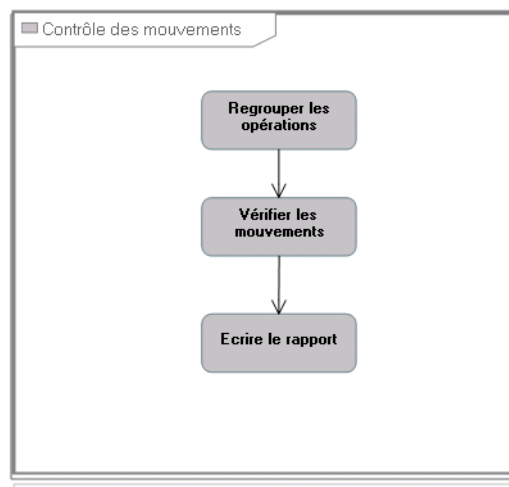
1. Cliquez sur le bouton **Séquence batch**.
2. Cliquez sur l'appel de traitement batch initial et, en maintenant enfoncé le bouton gauche de la souris, tirez un lien vers l'appel de traitement batch suivant.
3. Lâchez le bouton de la souris.
Le lien représentant l'ordonnancement des traitements apparaît dans le diagramme.

Créer un diagramme de structure de programme batch

 *Un programme est une étape élémentaire d'exécution d'un traitement de lot qui consiste à lancer l'exécution d'un programme avec les paramètres appropriés.*

Créer un diagramme de structure de programme batch

L'enchaînement de traitements d'un programme peut être décrit dans un **diagramme de structure de programme batch**.



Pour créer le diagramme de structure de programme batch:

1. Cliquez avec le bouton droit sur le programme qui vous intéresse.

2. Sélectionnez **Nouveau > Diagramme de structure de programme batch**.
Le diagramme s'ouvre.

Ajouter un appel de programme dans le diagramme

Les composants d'un **Programme** sont définis à partir des **Appels de programme** qui sont positionnés dans le diagramme.

Pour ajouter un composant dans le diagramme :

1. Sélectionnez le bouton **Appel de programme** et cliquez dans le diagramme.
La fenêtre **d'ajout d'un appel de programme** s'ouvre.
2. Cliquez sur la flèche à droite du champ **Traitement de lot appelé** et sélectionnez **Relier** dans la liste déroulante.
Une fenêtre de connexion apparaît.
3. Sélectionnez **Programme** dans la partie gauche de la fenêtre et cliquez sur la flèche de recherche.
La liste des plannings d'exploitation apparaît.
4. Sélectionnez le programme qui vous intéresse et cliquez sur le bouton **Relier**.
5. Cliquez sur **OK**.
L'appel de programme apparaît dans le diagramme.

L'ordre d'exécution des programmes est défini par des liens. Pour plus de détails, voir ["Définir les séquences des batchs"](#), page 93.

Décrire l'implémentation d'un service applicatif

L'exécution d'un service applicatif peut être décrite par un diagramme de structure de chaîne de traitement de lot. Pour plus de détails, voir ["Créer un diagramme de structure de chaîne de traitement de lot"](#), page 92.

Pour décrire qu'un service applicatif est implémenté par un planning d'exploitation, par exemple :

1. Ouvrez la fenêtre de propriétés du service applicatif qui vous intéresse.
2. Sélectionnez l'onglet **Implémentation**.
3. Dans la section **Service/Implémentation**, cliquez avec le bouton droit sur le dossier **Implémentation par traitement de lot** et sélectionnez **Nouveau**.
La fenêtre de création d'un service applicatif implémenté par un batch apparaît.
4. Cliquez sur la flèche à droite du champ **Traitement de lot implémenteur** et sélectionnez **Relier** dans la liste déroulante.
Une fenêtre de connexion apparaît.
5. Sélectionnez **Planning d'exploitation** dans la partie gauche de la fenêtre et cliquez sur la flèche de recherche.
La liste des planning d'exploitation apparaît.
6. Sélectionnez le planning qui vous intéresse et cliquez sur le bouton **Relier**.

7. Cliquez sur **OK**.
Le service applicatif implémenté par un batch apparaît dans la liste. Il est relié au planning d'exploitation sélectionné.

Utiliser les réalisations

Un mécanisme de réalisation permet d'associer un processus applicatif à un **Planning d'exploitation** ou par un **Programme**.

Pour décrire qu'un planning d'exploitation est associé à un processus applicatif :

1. Ouvrez la fenêtre de propriétés du planning d'exploitation qui vous intéresse.
2. Sélectionnez l'onglet **Caractéristiques** et le sous-onglet **Réalisation**.
3. Dans la section **Réalisation de composite**, cliquez avec le bouton **Nouveau**.
La fenêtre de création d'une réalisation batch de processus système apparaît.
4. En face du champ **Type de base**, sélectionnez **Relier**.
La fenêtre de recherche apparaît.
5. Sélectionnez le processus applicatif qui vous intéresse et cliquez sur **Relier**.
6. Cliquez sur **OK**.
La réalisation batch de processus système apparaît dans la page de propriétés du planning d'exploitation.

DÉCRIRE LES INTERFACES UTILISATEUR

Il est possible de décrire les interfaces reliant les services ou les opérations avec l'extérieur. Cette description s'effectue dans un diagramme d'IHM.

Créer une interface utilisateur

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Pour créer une interface utilisateur :

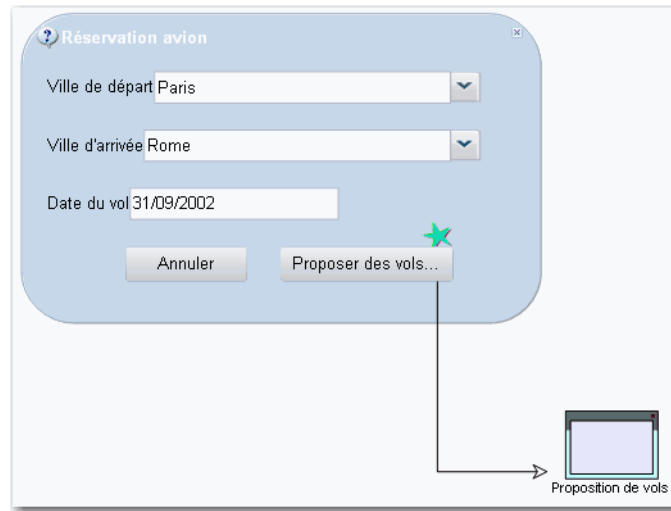
1. Cliquez sur le volet de navigation **Projet en cours**.
2. Dans le volet de navigation cliquez sur **Conception de l'architecture applicative** :
3. Cliquez sur la tuile **Interfaces utilisateur**.
La liste des interfaces utilisateur apparaît.
4. Cliquez sur **Nouveau**.
5. Saisissez le nom de l'interface.
6. Cliquez sur **OK**.

Construire un diagramme d'IHM

Pour créer un diagramme d'IHM :

1. Faites un clic droit sur l'interface utilisateur qui vous intéresse.
2. Sélectionnez **Nouveau > Diagramme d'IHM**.
Le diagramme d'IHM s'ouvre dans la fenêtre d'édition.

Prenons l'exemple du diagramme d'IHM du service "Réservation Avion".



L'interface se présente sous forme d'une boîte de dialogue, dans laquelle doivent être renseignés différents champs :

- La ville de départ
- La ville d'arrivée
- La date du vol

Un bouton annule la demande, un autre ouvre une seconde interface.

Dessiner l'IHM

Le diagramme d'IHM vous permet de dessiner l'interface de l'opération ou du service.

Élément D'IHM

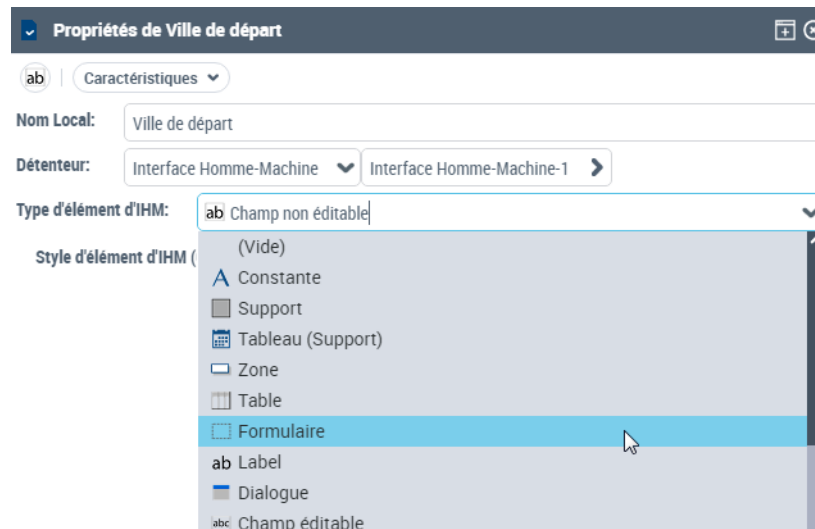
Un certain nombre de boutons vous permettent de définir l'apparence de l'interface :

- Champ de texte
- Liste
- Bouton radio
- Case à cocher
- Etc.

Pour créer un élément :

1. Dans la barre d'objets du diagramme, sélectionnez le bouton correspondant à l'élément voulu puis cliquez dans le diagramme.
2. Dans la fenêtre qui apparaît, tapez le nom de l'élément.
3. Cliquez sur **OK**.

Vous pouvez également cliquer sur le bouton **Élément d'IHM** et indiquer le type de l'élément dans sa fenêtre de propriétés.



Événement d'IHM

Vous pouvez relier un *événement* à un élément d'IHM. Dans notre exemple, le bouton "Proposer des vols" est relié à un événement, qui, lorsqu'il est actionné, ouvre une autre interface.

Pour créer un événement :

1. Cliquez sur le bouton **Événement d'IHM** puis cliquez dans le diagramme.
2. Saisissez le nom de l'événement et cliquez sur **OK**.

Type d'événement

Il existe différents types d'événement. Il peut s'agir :

- D'un clic sur un bouton
- D'une entrée dans un champ
- Etc.

Pour spécifier le type d'un événement :

1. Ouvrez la fenêtre de propriétés de l'événement.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Dans le champ **Événement type d'IHM**, cliquez sur la flèche puis sur **Rechercher Événement type d'IHM**. La fenêtre de sélection apparaît.
4. Cliquez sur **Chercher**. La liste des types d'événement apparaît.
5. Sélectionnez le type voulu et cliquez sur **OK**.

Relier l'événement à un élément

Pour relier l'événement à un élément, deux possibilités s'offrent à vous :

- Sélectionnez l'événement dans le diagramme et faites-le glisser sur l'élément.
- Ou ouvrez la fenêtre de propriétés de l'événement et renseignez le champ **Élément d'IHM**.

LES RAPPORTS SYSTÈME DE HOPEX APPLICATION DESIGN

Deux rapports MS Word sont disponibles dans le cadre de la spécification technique d'un projet avec **HOPEX Application Design**.

Lancer un rapport système (MS Word) sur un projet

Pour lancer un rapport système d'un projet de spécification (AD) :

1. Cliquez sur le menu de navigation puis sur **Projets en cours**.

 L'accès au volet de navigation **Projet en cours** nécessite que vous ayez défini un environnement de travail sur le projet concerné. Voir "[Le volet Projet en cours](#)", page 22.

2. Dans le volet de navigation cliquez sur **Conception de l'architecture applicative**.
3. Dans la zone d'édition cliquez sur la tuile du rapport voulu.
4. Cliquez sur **Nouveau**.
L'assistant de création du rapport apparaît.
5. Indiquez le nom du rapport.
6. Cliquez sur **Suivant** puis sur **OK**.

Pour ouvrir le document généré :

1. Cliquez sur l'icône du document et sélectionnez **Documentation > Ouvrir**.

Rapport d'analyse du système

Ce rapport type porte plus précisément sur les impacts du système étudié.

Le rapport généré présente :

- tous les diagrammes d'impact contenus dans les paquetages du projet, avec le détail des éléments influencés.
- les diagrammes de scénarios d'interaction et les diagrammes de séquence contenus dans les paquetages de cas d'utilisation du projet. Les cas d'utilisation doivent être "utilisés" par le projet (voir "[Etat des livrables du projet](#)", page 172).
- le détail des applications et des services "Utilisés" dans le projet.
- les modèles de données "utilisés" dans le projet.
- la matrice de traçabilité des exigences.

Rapport de conception du système

Ce rapport type permet de décrire la structure interne d'une application du projet, les services IT ainsi que les bases de données de cette application.

La description des services concerne les services "produits" ou "modifiés" au sein du projet. Il s'agit des services appartenant à l'architecture interne de l'application analysée, autrement dit reliés à l'application par le lien "service-défini".

Chaque rapport porte sur une application en particulier. A partir de ce rapport type, vous pouvez générer autant de rapports qu'il y a d'applications.

Un rapport de type "Système Blueprint - Conception du système" présente :

- l'arbre applicatif de l'application décrite.
- les diagrammes d'architecture interne de l'application.
- les modèles de données qui décrivent les bases de données de l'application.
- les processus applicatifs des services.
- les diagrammes de scénarios d'interaction et les diagrammes de séquence des cas d'utilisation qui implémentent les services de l'application.
- la matrice de traçabilité des exigences.

UTILISER LES DIAGRAMMES D'ARCHITECTURE INFORMATIQUE



Ce chapitre présente comment construire les principaux types de diagramme d'architecture IT.

Le tableau ci-dessous donne la liste des solutions pour lesquelles les différentes descriptions sont proposées pour les différents types d'objet.


Type d'objet	Diagramme de structure	Scénario de flux	Architecture technique
Application	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture
Système applicatif	HOPEX IT Architecture	HOPEX IT Architecture	HOPEX IT Architecture
Environnement d'application	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture
Service applicatif	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture
Micro-Service	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture	HOPEX Application Design HOPEX IT Architecture

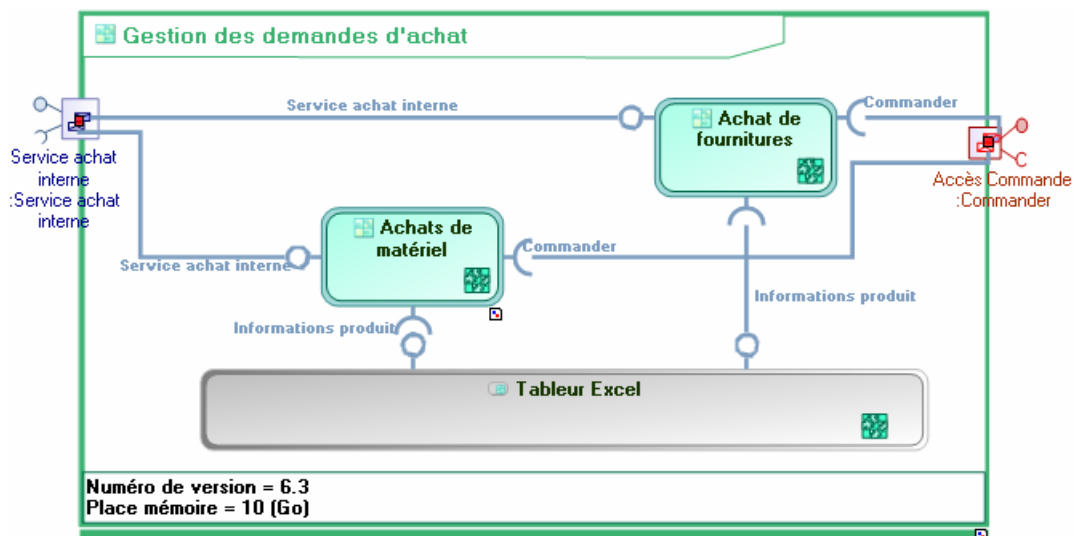
- ✓ ["Créer un diagramme de structure", page 152 ;](#)
- ✓ ["Décrire les scénarios de flux", page 154 ;](#)
- ✓ ["Décrire une architecture technique", page 160.](#)

CRÉER UN DIAGRAMME DE STRUCTURE

Avec **HOPEX IT Architecture**, les composants d'un objet et leurs échanges sont décrits dans un diagramme de structure.

Créer le diagramme de structure d'application

 Un diagramme de structure d'application montre graphiquement les composants de premier niveau d'une application, les points d'accès (point de service et de requête) et les connexions entre les composants.




L'application de gestion des demandes d'achats, qui n'est utilisée que pour les achats internes, s'appuie sur deux services applicatifs spécialisés : l'un dans l'achat des fournitures et l'autre dans l'achat de matériel. Les deux services applicatifs utilisent Microsoft Excel.

Les composants d'un diagramme de structure d'application

Un diagramme de structure d'application est composé de :


- **services applicatifs** qui représentent les services applicatifs utilisés et déployés avec l'application.

Dans l'exemple, il s'agit du service d'achat des fournitures.

 *Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.*

- **micro-services** qui représentent les services utilisés indépendants de l'application.


Dans l'exemple, il s'agit de l'application Microsoft Excel.

 *Un micro-service est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.*


- points d'accès, de requête et de service.

 Pour plus de détails, voir ["Décrire les points de service et de requête", page 165](#).

- **interactions** entre les composants.

 *Une interaction représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par un contrat d'échange.*

- **dépôts de données physiques** utilisés par l'application.

 Pour plus de détails, voir ["Gérer les données", page 73](#).

Ajouter un service applicatif dans un diagramme de structure d'application

Pour décrire qu'une application utilise un service applicatif, vous allez :

1. Dans la barre d'objets du diagramme de structure d'application, cliquez sur le bouton **Service applicatif** et cliquez dans le cadre de l'application décrite.
Une fenêtre d'ajout vous demande de choisir le **Service applicatif** utilisé.
2. Sélectionnez un service applicatif existante
3. Cliquez sur **OK**.
Le service applicatif apparaît dans le diagramme.

DÉCRIRE LES SCÉNARIOS DE FLUX

Un diagramme de scénario de flux décrit les messages échangés entre les éléments du système représenté.

Avec **HOPEX**, deux types de diagramme sont proposés.

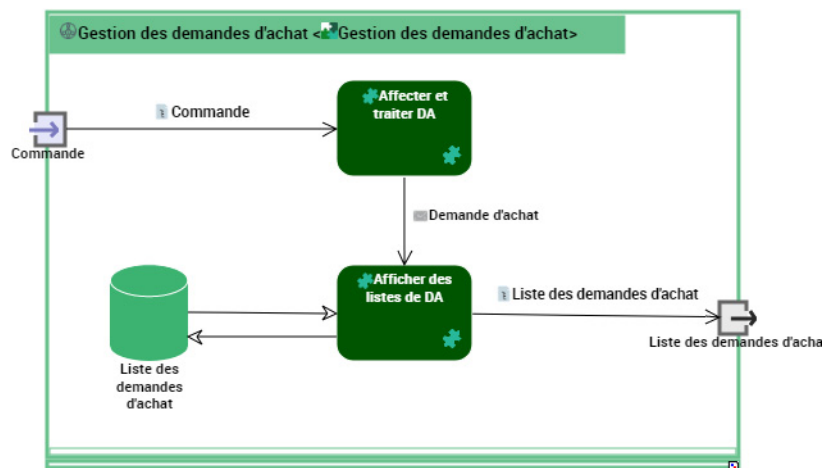
- Les *diagrammes de scénario de flux* qui décrivent les flux échangés dans différents scénarios d'utilisation de l'objet décrit.
- Les *diagrammes de séquences de scénario de flux* qui décrivent la chronologie des flux échangés dans différents scénarios d'utilisation de l'objet décrit.

Utiliser un diagramme de scénario de flux d'application

Un diagramme de scénario de flux peut être construit pour un environnement d'application, une application, un système applicatif, un service applicatif ou un micro-service.

Le diagramme de scénario de flux ci-après décrit l'environnement de l'application "Achat de pièces détachées".

Le scénario de flux d'application ci-dessous décrit les échanges entre les composants de l'application "Gestion des demandes d'Achat".



Exemple de scénario de flux d'application de "Gestion des demandes d'achat"

Dans un diagramme de scénario de flux d'application, les éléments représentés sont :

- des services applicatifs,
- des micros services,
- des dépôts de données applicatives internes ou externes,
- des ports applicatifs en entrée ou en sortie.

Les échanges proposés entre ces éléments sont :


- des flux applicatifs qui portent un contenu,
- des canaux de flux applicatifs qui regroupent plusieurs flux applicatifs sur un même lien,
- des canaux de données applicatives qui représentent les échanges effectués avec les dépôts de données applicatives.

Créer un diagramme de scénario de flux d'application

Pour créer un scénario de flux d'application :

1. Faites un clic droit sur l'application et cliquez sur **Nouveau > Scénario de flux d'application**.
2. Dans la fenêtre de choix du type de diagramme, sélectionnez **Scénario de flux d'application**.


Ajouter un service applicatif au scénario de flux d'application

 *Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.*

Pour ajouter un **service applicatif** :


1. Dans la barre d'objets du scénario de flux d'application, cliquez sur le bouton **Service applicatif**.
2. Cliquez dans le cadre de l'application décrite.
Une fenêtre d'ajout vous demande de choisir le **Service applicatif** que vous souhaitez utiliser (par exemple "Gestion des clients").
3. Sélectionnez le service applicatif et cliquez sur **OK**.
Le service applicatif apparaît dans le diagramme.

De la même manière vous pouvez ajouter un micro-service.

 *Un micro-service est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.*

Gérer les flux applicatifs dans un scénario de flux d'application

Créer un flux applicatif avec un contenu

 *Un flux applicatif représente la circulation d'informations entre des applications ou à l'intérieur d'une application. Un flux applicatif peut transporter un contenu.*

Les flux applicatifs échangés entre les services applicatifs, les micro-services ou les ports applicatifs d'un scénario de flux d'application sont associés à un **contenu**.



Le contenu désigne le contenu d'un message ou d'un événement indépendamment de sa structure. Cette dernière peut être représentée par un schéma XML relié au contenu. Un contenu peut être utilisé par plusieurs messages puisqu'il n'est pas associé à un émetteur et à un destinataire. Il ne peut y avoir qu'un contenu par message ou par événement, mais un même contenu peut être utilisé par plusieurs messages ou événements.

Vous devez préciser le **contenu** d'un **flux applicatif** directement lors de sa création.

Pour créer le **flux applicatif** :

1. Cliquez sur la bobine de la barre d'objets d'un scénario de flux d'application.
2. Cliquez sur le premier objet qui représente l'émetteur du flux et, en maintenant le bouton de la souris enfoncé, tirez un trait jusqu'à l'objet récepteur du flux.
La fenêtre de **Création de Flux applicatif** s'ouvre.
3. Sélectionnez le contenu que vous souhaitez associer au flux à partir de la liste déroulante du champ **Contenu**.
Le flux applicatif s'affiche avec son contenu dans le diagramme.

Créer un canal de flux applicatifs



Un canal de flux applicatif permet de regrouper graphiquement plusieurs flux applicatifs en un seul flux.

Pour créer un canal de flux applicatif, vous devez d'abord créer le canal et relier ensuite les flux applicatifs qu'il regroupe.

Pour créer un **canal de flux applicatifs** :

1. Dans la barre d'objets du scénario de flux d'application, cliquez sur le bouton **canal de flux applicatifs**.
2. Cliquez sur le premier objet en communication et, en maintenant le bouton de la souris enfoncé, tirez un trait jusqu'à l'autre objet.
Le canal de flux applicatif apparaît dans le diagramme.

Pour relier les flux applicatifs au **canal de flux applicatifs** :

1. Ouvrez la page de propriétés **Caractéristiques** du canal de flux applicatif.
2. Dans la section **Flux Groupés**, cliquez sur le bouton **Relier**.
Une fenêtre de sélection s'ouvre et présente la liste des flux applicatifs non groupés du scénario de flux d'application.
3. Sélectionnez les flux que vous souhaitez regrouper et cliquez sur **OK**.
Les contenus des flux applicatifs s'affichent avec une flèche qui marque le sens du flux.

Ajouter un dépôt de données applicatives au scénario de flux d'application



Un dépôt de données applicatives matérialise l'utilisation de données dans le contexte d'un composant logiciel (par exemple une application). Un dépôt de données applicatives fournit un mécanisme pour mettre à jour ou consulter des données stockées à l'intérieur du composant logiciel en cours.





Pour plus de détails sur la gestion des dépôts de données, voir "Gérer les données", page 73.

Un dépôt de données peut être local ou externe à l'application.

Pour ajouter, par exemple, un dépôt de données applicatives locales dans un scénario de flux d'application :

1. Dans la barre d'objets du scénario, cliquez sur le bouton **Dépôt de données applicatives local**.
2. Cliquez dans le cadre de l'application décrite.
Une fenêtre d'ajout vous demande de choisir le *domaine de données* qui représente la structure physique qui va supporter concrètement le dépôt de données applicatives.

 *Un domaine de données représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles. Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.*

 *Pour plus de détails sur les domaines de données, voir le chapitre "Les domaines de données logiques et applicatifs" du guide **HOPEX Information Architecture**.*


3. Sélectionnez le **Domaine de données** existant qui vous intéresse.
4. Cliquez sur **OK**.
Le dépôt de données applicatives local apparaît dans le diagramme, il porte le nom du domaine de données physique sélectionné.

Créer un canal de données applicatives

Les applications, les systèmes applicatifs et les micro-services peuvent accéder en lecture ou en écriture à un dépôt de données applicatives local ou externe.

Pour créer un canal de données applicatives qui représente un accès en lecture :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Canal de données applicatives**.
2. Tirez un lien entre le dépôt de données applicatives et l'objet qui lit les données.
Un canal de données applicatives est automatiquement apparaît dans le scénario.

 *Pour créer un lien en écriture, vous devez tirer un lien entre l'objet qui écrit et le dépôt de données applicatives.*

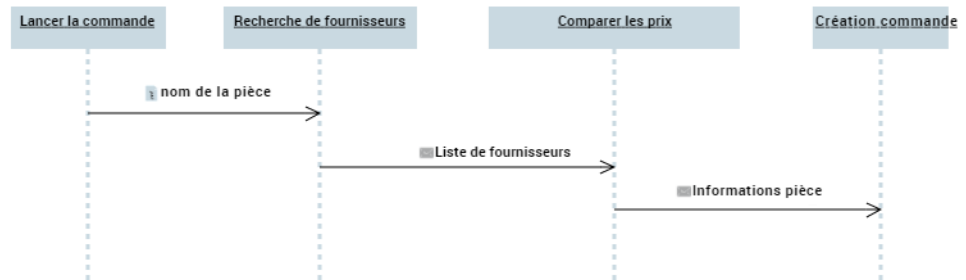
Utiliser un diagramme de séquence de scénario de flux

Pour chaque contexte d'utilisation, vous pouvez créer des diagrammes de séquence de scénario de flux. Un diagramme de séquence de scénario de flux présente les mêmes échanges entre les éléments d'un système, en mettant en évidence leur chronologie. Les éléments du scénario sont représentés dans le diagramme par des lignes.

Un diagramme de séquence de scénario de flux contient :

- Des lignes qui définissent les participants de l'interaction tels que les instances d'applications, de services ou d'IHM.
- Différents types de messages échangés entre les participants.
- Des fonctionnalités avancées qui permettent de décrire de manière concise plusieurs séquences d'exécution.

☛ Pour plus de détails sur les objets d'un diagramme de séquence, voir le guide **HOPEX UML**.



Ce diagramme décrit le fonctionnement du cas d'utilisation "Commander des pièces non référencées" :

Quand une demande d'achat est saisie dans l'interface utilisateur, le nom de la pièce est reçu par le service "Recherche de fournisseurs", qui dresse la liste des fournisseurs proposant la pièce demandée.

Le service "Comparer les prix" recherche le produit le moins cher et envoie les informations à un service qui calcule le montant de la commande.

Une fois le montant établi, un dernier service passe la commande auprès de l'interface.

Créer un diagramme de séquence de scénario de flux

Pour créer un scénario de séquence de flux d'environnement d'application, par exemple :

1. Faites un clic droit sur l'environnement d'application et cliquez sur **Nouveau > Scénario de flux d'environnement d'application**.
2. Dans la fenêtre de choix du type de diagramme, sélectionnez **Séquence de scénario de flux d'environnement d'application**.

Instances d'application, de service applicatif ou d'IHM

Selon qu'il porte sur une IHM, une application ou un service applicatif, le diagramme de scénario d'interaction décrit les messages échangés entre des instances d'application, des instances de **service applicatif** ou des instances d'**IHM**.

📖 Une interface homme-machine permet de définir un écran d'une application ou d'un service applicatif.

📖 Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des

fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.

Pour créer une instance de service applicatif, par exemple :

1. Cliquez sur le bouton **Service applicatif** de la barre d'outils.
2. Cliquez dans le diagramme.
Une boîte d'**Ajout de service applicatif** s'affiche.
3. Cliquez sur la flèche à droite du champ **Nom** et sélectionnez **Relier Service applicatif** dans la liste déroulante.
La liste des services applicatifs accessibles à partir de la bibliothèque courante apparaît.
4. Sélectionnez le service applicatif de votre choix.
5. Cliquez sur le bouton **OK**.
L'instance de service applicatif apparaît dans le diagramme.

Instances de messages

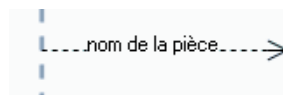
Les instances de messages définissent les informations qui circulent entre les instances d'applications, de services applicatifs ou d'IHM. La séquence décrite dans le diagramme de séquence de scénario de flux indique l'ordre d'envoi des messages.

Les instances de messages représentées dans le diagramme de scénario d'interaction correspondent aux messages détenus par l'application et définis précédemment dans un autre diagramme.

Pour créer une instance de message :

1. Cliquez sur la bobine de la barre d'outils.
2. Cliquez sur la ligne pointillée sous le premier objet et, en maintenant le bouton gauche de la souris enfoncé, tracez une ligne jusqu'au deuxième objet.
3. Relâchez le bouton gauche de la souris.

L'instance de message échangée entre les deux objets se dessine.

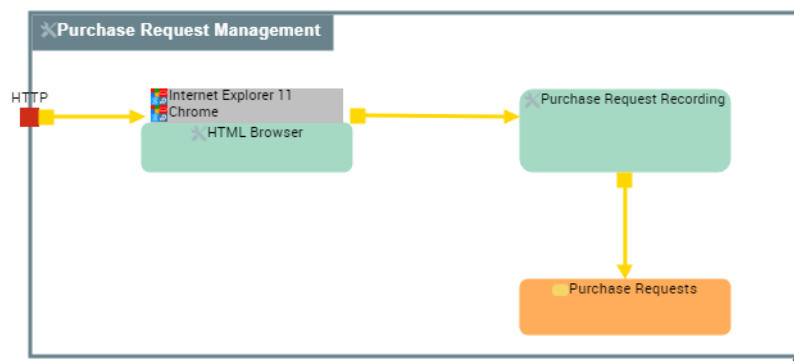


DÉCRIRE UNE ARCHITECTURE TECHNIQUE

Une *architecture technique* permet de représenter les architectures techniques et les zones techniques de données de l'objet décrit ainsi que les techniques utilisées pour leurs communications.

L'architecture technique d'application ci-dessous présente les zones techniques utilisées par l'application de gestion des demandes d'achat.

La ligne de communication technique s'appuie sur http.



Exemple d'architecture technique



Une architecture technique d'application décrit une des configurations possibles de déploiement d'une application. Elle décrit comment les différentes zones techniques de l'application sont reliées entre elles et les technologies et les protocoles de communication qu'elles utilisent. Une application peut avoir plusieurs architectures techniques possibles (Ex: installation autonome, déploiement horizontal ou vertical, etc.)

Créer un diagramme d'architecture technique d'application

Les éléments représentés dans un diagramme d'architecture technique d'application sont :

- des *zones techniques d'application*,



Une zone technique d'application représente un élément du découpage d'une application en fonction de critères techniques. Par exemple, ce peut être l'interface utilisateur ou un traitement. Chaque zone technique d'application est associée à une ou plusieurs technologies. Le déploiement de plusieurs zones techniques d'application est nécessaire pour que l'application soit opérationnelle.

- des *zones techniques de données*,



Une zone technique de données représente un élément du découpage d'une application permettant l'accès aux données nécessaires au fonctionnement de cette application. Chaque zone

technique de données est associée à une ou plusieurs technologies (Ex: Oracle 12, SQL Server 2012, ...). Une zone technique de données peut permettre l'accès à un ou plusieurs dépôts de données.

- des **ports techniques en entrée** et en sortie,



Un port technique serveur est un point permettant l'ouverture d'une communication vers une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).

- des **lignes de communication techniques**.



Une ligne de communication technique représente une connexion technique entre des architectures ou des zones techniques d'application par le biais de ports techniques client ou serveur. Le port technique client d'une architecture ou d'une zone technique demande l'ouverture de la ligne de communication au port technique serveur de l'autre zone ou architecture technique.

Ajouter une zone technique d'application à un diagramme d'architecture technique d'application



Une zone technique d'application représente un élément du découpage d'une application en fonction de critères techniques. Par exemple, ce peut être l'interface utilisateur ou un traitement. Chaque zone technique d'application est associée à une ou plusieurs technologies. Le déploiement de plusieurs zones techniques d'application est nécessaire pour que l'application soit opérationnelle.

Pour ajouter une **zone technique d'application** :

1. Dans la barre d'objets de l'architecture technique d'application, cliquez sur le bouton **Zone technique d'application**.
2. Cliquez dans le cadre de l'application décrite.
Une fenêtre d'ajout vous demande de choisir la **Zone technique d'application** que vous souhaitez utiliser.
3. Sélectionnez la zone technique d'application et cliquez sur **OK**.
La zone technique d'application apparaît dans le diagramme.

De la même manière vous pouvez ajouter des zones techniques de données.



Une zone technique de données représente un élément du découpage d'une application permettant l'accès aux données nécessaires au fonctionnement de cette application. Chaque zone technique de données est associée à une ou plusieurs technologies (Ex: Oracle 12, SQL Server 2012, ...). Une zone technique de données peut permettre l'accès à un ou plusieurs dépôts de données.

Définir les technologies logicielles utilisées par une zone technique d'application



Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.

Pour spécifier les **technologies logicielles requises** pour une **zone technique d'application** :

1. Ouvrez la page de propriétés **Caractéristiques** de la **Zone technique d'application** qui vous intéresse.

2. Dans la section les **technologies logicielles requises**, cliquez sur **Relier**.

Une fenêtre de sélection vous permet de sélectionner les **Technologies logicielles** que vous souhaitez utiliser.

Les technologies logicielles sélectionnées apparaissent sur l'icône de la zone technique d'application.

Créer une ligne de communication technique



Une ligne de communication technique représente une connexion technique entre des architectures ou des zones techniques d'application par le biais de ports techniques client ou serveur. Le port technique client d'une architecture ou d'une zone technique demande l'ouverture de la ligne de communication au port technique serveur de l'autre zone ou architecture technique.

Les techniques des communication entre les zones techniques d'application et de données peuvent être décrites par des lignes de communication technique.

Pour créer une lignes de communication technique, vous devez d'abord créer la ligne et spécifier ensuite les protocoles de communication qui sont utilisés.

Pour créer une ligne de communication technique :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton Ligne de communication technique.
2. Tirez un lien entre les deux objets en communication.
La ligne de communication technique apparaît dans l'architecture.

Pour relier les protocoles de communication à la **ligne de communication technique** :

1. Ouvrez la page de propriétés **Caractéristiques** de la ligne de communication technique.
2. Cliquez sur le bouton **Relier**.
Une fenêtre de sélection s'ouvre et présente la liste des formats de communication.
3. Sélectionnez les formats que vous souhaitez utiliser et cliquez sur **Relier**.

DÉCRIRE LES ÉCHANGES DE DONNÉES




Ce chapitre présente comment décrire les contrats d'échange entre les composants d'une l'architecture métier ou informatique.


- ✓ ["Gérer les interactions", page 164 ;](#)
- ✓ ["Décrire un échange", page 169 ;](#)
- ✓ ["Décrire un contrat d'échange", page 172.](#)


GÉRER LES INTERACTIONS

Une *interaction* représente l'échange d'informations entre des composants d'architecture.

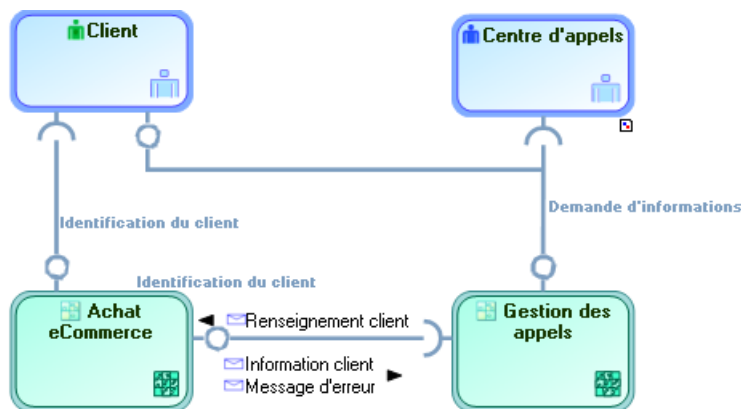
 Une interaction représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par un contrat d'échange.

Le contenu d'une interaction est décrit par un *contrat d'échange*.

 Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.

 Pour plus de détails sur les notions de contrat d'échange, voir ["Décrire un contrat d'échange", page 172](#).

Dans un diagramme de structure de système applicatif de "Traitement des demandes d'achat", deux contrats d'échanges sont utilisés par les différentes interactions.





Interactions dans le diagramme de structure de système applicatif "Traitement des demandes d'achat"

Les clients doivent être identifiés avant de passer une commande. Ils peuvent passer des commandes directement à partir d'une application de eCommerce ou en passant par un Centre d'appels. Le centre d'appels passe par l'application de "Gestion des appels" qui utilise le service d'identification du client proposé par l'application "Achat eCommerce".

Créer une interaction

Pour créer une interaction :


1. Dans la barre d'objets d'un diagramme, cliquez sur le bouton **Interaction** .
2. Cliquez sur l'entité qui demande le service et tirez un lien l'entité qui fourni le service.
3. Dans la fenêtre d'ajout d'interaction, précisez le contrat d'échange que vous souhaitez utiliser.


 Vous pouvez également créer un nouveau contrat d'échange. Pour plus de détails, voir "[Créer un contrat d'échange à partir d'une interaction](#)", page 174.

4. Cliquez sur **Ajouter**.

Décrire les points de service et de requête

Dans une architecture orientée service, les communications s'appuient sur des points d'accès : *points de service* et des *points de requête*.


 Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.


 Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.

Les points de service

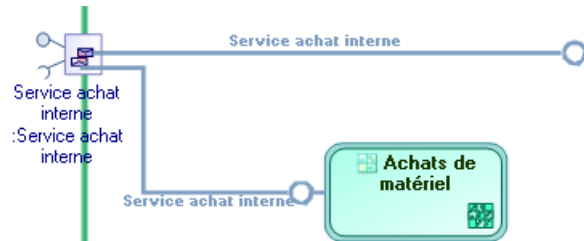
Un système applicatif, par exemple, est créé pour assurer un ou plusieurs services. Ces services sont représentés par des *points de service*.

Le service est demandé selon des modalités précises définies par un *contrat d'échange* affecté au point de service.

 Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.

 Pour plus de détails sur les contrats d'échange, voir "[Décrire un contrat d'échange](#)", page 172.


Les composants activés pour assurer un service sont liés au point de service par des interactions. S'il est nécessaire d'activer plusieurs composants, vous devez créer plusieurs interactions entre le point de service et les composants du système.




Dans l'exemple présenté ici, le service d'achat interne est lié à deux interactions, basées sur le même contrat d'échange, qui représentent l'activation de l'application d'achat de matériel ou l'application d'achat de fournitures.


➡ Pour créer un point de service, voir "[Créer un point de service ou un point de requête](#)", page 167.

Les points de requête

Un **point de requête**  permet de représenter l'utilisation d'un service externe à l'entité décrite.

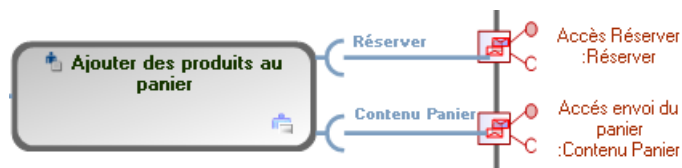
 Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.

Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels. Le service est demandé selon des modalités précises définies par un **contrat d'échange** affecté au point de requête.

 Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.

➡ Pour plus de détails sur les contrats d'échange, voir "[Décrire un contrat d'échange](#)", page 172.

Les composants qui émettent une requête sont liés au point de requête par une interaction.



Dans l'exemple, les points de requête représentent les demandes de service effectuées par l'utilisation d'une

application logique "Ajouter des produits au panier" pour passer une commande ou réserver les produits.

☛ Pour créer un point de requête, voir "[Créer un point de service ou un point de requête](#)", page 167.


Créer un point de service ou un point de requête

Le processus de création d'un **point de service** ou d'un **point de requête** est identique.

📖 Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.

📖 Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.

Pour créer un point de service :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Point de service** .
2. Positionnez l'objet sur le bord du cadre de l'objet décrit. Une fenêtre de création s'ouvre.
3. Cliquez sur la flèche à droite du champ **Contrat d'échange** pour définir le contrat d'échange qui permet l'activation de ce point de service et sélectionnez, par exemple, **Relier Contrat d'échange**. La fenêtre de recherche s'ouvre.
4. Sélectionnez le contrat d'échange qui est associé à ce point de service et cliquez sur **Relier**.
5. Cliquez sur **Suivant**. Une nouvelle fenêtre s'ouvre pour vous proposer la liste des rôles du contrat d'échange pouvant être associés au point de service.

☛ Cette fenêtre n'est pas proposée s'il n'y a qu'un seul rôle candidat à être associé au point de service.
6. Sélectionnez le rôle qui vous intéresse et cliquez sur **OK**. Le point de service apparaît dans le diagramme.

Pour changer le nom du point de service :

1. Cliquez sur le nom du point de service et appuyez sur la touche <F2>.
2. Saisissez le nouveau nom utilisé lors de la spécification des points d'interaction.

☛ Pour plus de détails sur les points d'interactions, voir "[Décrire les points de service et de requête](#)", page 165.



Définir le point d'interaction d'un élément

Le point d'interaction d'un élément relie une interaction à l'un des composants en communication. Il permet de spécifier :

- d'une part, le point de service, ou le point de requête, qui intervient dans la communication
- d'autre part, le rôle, consommateur ou fournisseur, que le point d'interaction représente dans le contrat d'échange.

Caractériser le point d'interaction d'un élément

Pour modifier les propriétés du point d'interaction d'un élément :

1. Faites un clic droit sur l'interaction à proximité de l'élément en communication.
2. Ouvrez la page de propriétés **Caractéristiques**.
3. Sélectionnez la **Participation à un rôle de service**, c'est-à-dire le rôle du contrat d'échange joué par le point d'interaction de l'élément.
 Pour plus de détails sur les rôles d'un contrat d'échange, voir "[Créer un diagramme d'échange \(BPMN\)](#)", page 170.
4. Sélectionnez le **Point d'interaction cible**, c'est-à-dire le point de service (ou de requête) qui est concerné par l'interaction.
 Pour plus de détails sur les points de service ou de requête, voir "[Décrire les points de service et de requête](#)", page 165.
5. Cliquez sur **OK**.

DÉCRIRE UN ÉCHANGE

Le contenu d'une interaction est décrit par un *contrat d'échange*.

Un *contrat d'échange* est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.

Pour plus de détails sur les contrats d'échange, voir "*Décrire un contrat d'échange*", page 172.

Un contrat d'échange peut être décrit par un enchaînement d'échanges ou de contrats d'échange.

Un *échange* décrit les échanges de flux entre les participants.

Un diagramme d'échange décrit l'enchaînement des flux d'un *échange*.

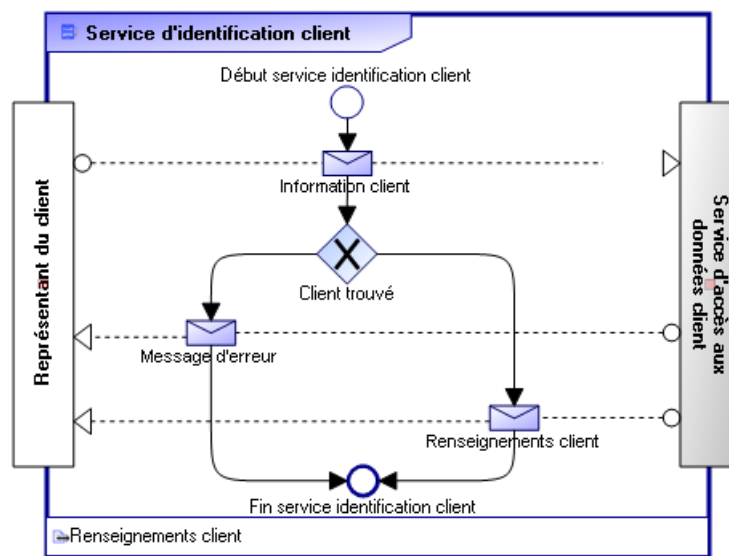


Diagramme d'échange de "Service d'identification client"

Le déroulement du service d'identification de client commence par l'envoi d'information permettant d'identifier le client. Un message d'erreur est émis si le client n'a pas été trouvé, sinon les renseignements client sont envoyés (identification client, état des commandes, etc.).

Créer un échange

Vous pouvez créer un *échange* à partir d'un diagramme de contrat d'échange (BPMN).

Pour créer un *échange* à partir d'un diagramme de contrat d'échange (BPMN) :

1. Sélectionnez le bouton **Echange utilisé**  et cliquez dans le diagramme à l'intérieur de la forme du contrat d'échange décrit.



Un échange utilisé représente l'utilisation d'un échange dans un contrat d'échange.

La fenêtre de création d'échange utilisé s'ouvre.

2. Cliquez sur la flèche à droite du champ **Spécification de l'échange** et sélectionnez **Créer Echange** dans la liste déroulante.
La fenêtre de création d'un échange s'ouvre.
3. Saisissez le **Nom** de votre échange et cliquez sur **OK**.
4. Sélectionnez dans le champ **A partir de** le rôle du contrat d'échange décrit relié au rôle "Consommateur" de l'échange utilisé.
5. Sélectionnez dans le champ **Jusqu'à** le rôle du contrat d'échange décrit relié au rôle "Fournisseur" de l'échange utilisé.
6. Cliquez sur **Terminer**.
7. Cliquez sur **OK**.
L'échange est automatiquement créé.

Décrire un échange

Créer un diagramme d'échange (BPMN)

Un *échange* est décrit par un diagramme d'échange qui présente l'enchaînement des messages échangés.

Pour créer un diagramme d'échange :

1. Cliquez avec le bouton droit sur un **Echange** et cliquez sur **Nouveau > Diagramme d'échange (BPMN)**.
Le diagramme s'ouvre. Le cadre de l'échange est positionné et les deux rôles (Consommateur et Fournisseur) sont créés.

Créer un flux avec contenu

Vous devez préciser les *flux* et leur *contenu* qui sont échangés entre les deux rôles de l'échange.



Un flux représente la circulation d'information à l'intérieur d'un contrat échange. Un flux transporte son contenu.



Le contenu désigne le contenu d'un message ou d'un événement indépendamment de sa structure. Cette dernière peut être représentée par un schéma XML relié au contenu. Un contenu peut être utilisé par plusieurs messages puisqu'il n'est pas associé à un émetteur et à un destinataire. Il ne peut y avoir qu'un contenu par message ou par événement, mais un même contenu peut être utilisé par plusieurs messages ou événements.


Pour créer le flux et son contenu :

1. Dans le diagramme d'échange, cliquez sur le bouton **Flux avec contenu**.


2. Cliquez sur le rôle qui représente l'émetteur du flux et tirez un trait jusqu'au rôle qui représente le récepteur du flux en maintenant le bouton de la souris enfoncé.
La fenêtre de **Création de Flux avec contenu** s'ouvre.
3. Sélectionnez le contenu que vous souhaitez associer au flux à partir de la liste déroulante du champ **Contenu**.
Le flux s'affiche avec son contenu dans le diagramme.

Gérer les événements, les branchements et les enchaînements


Les **événements** "Début" et "Fin" sont nécessaires à la description du service assuré par le contrat d'échange.


 *Un événement représente un fait ou une action se produisant dans le système, par exemple - modification de l'adresse client. Il est géré par un broker. Pour signaler qu'elle peut produire un événement, une application déclare qu'elle le publie. Si elle est intéressée par un événement, une application déclare qu'elle y souscrit.*

Conformément à la norme BPMN, dans la barre d'objets, plusieurs types de **branchement** vous sont proposés.

 *Un branchement est un élément de modélisation utilisé pour contrôler des enchaînements qui convergent ou qui divergent à l'intérieur d'un processus.*

Un **enchaînement** est un lien orienté qui représente l'organisation chronologique des différentes étapes de traitement.

 *Un enchaînement est utilisé pour montrer l'ordre d'exécution des étapes d'un contrat d'échange. Un enchaînement n'a qu'une source et une cible.*

 *Pour plus de détails sur les événements, branchements et enchaînements, voir ["Gérer les événements, les branchements et les enchaînements"](#), page 171.*

DÉCRIRE UN CONTRAT D'ÉCHANGE

Une *interaction* représente l'échange d'informations entre les composants de l'architecture.



Une interaction représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par un contrat d'échange.

Le contenu d'une interaction est décrit par un *contrat d'échange*.



Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.

Un contrat d'échange est décrit comme un enchaînement d'opérations qui sont représentées :

- soit par des *contrats d'échange utilisés*



Un contrat d'échange utilisé est associée à un contrat d'échange. Il permet de représenter des échanges complexes.

- soit par des *échanges utilisés*



Un échange utilisé représente l'utilisation d'un échange dans un contrat d'échange.



Pour plus de détails sur les échanges, voir "[Décrire un échange](#)", page 169.

Exemples de diagrammes de contrat d'échange (BPMN)

Diagramme de contrat d'échange (BPMN)

Le diagramme de contrat d'échange associé au contrat d'échange "Identification du client" décrit, dans un formalisme BPMN, les opérations réalisées.

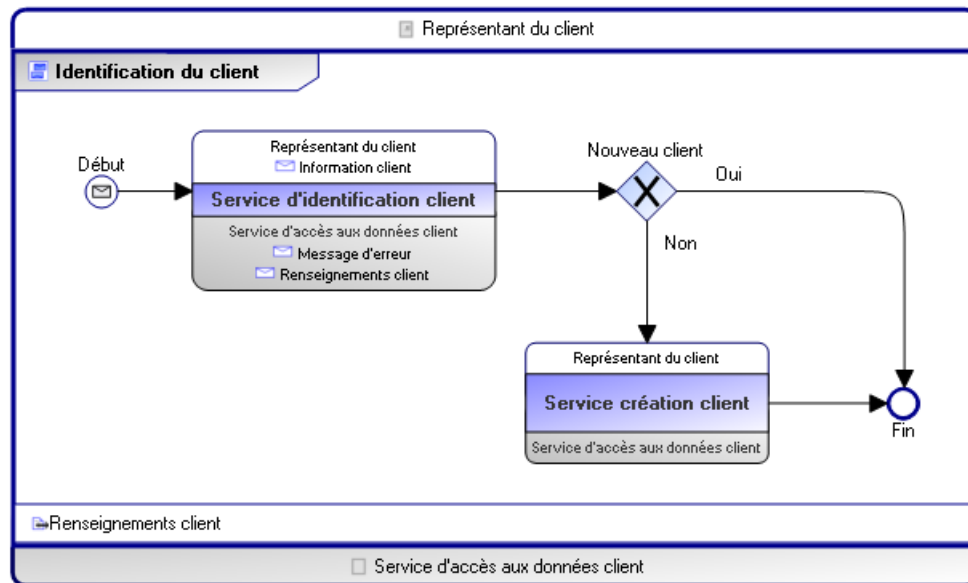



Diagramme de contrat d'échange (BPMN) "Identification du client"

Le déroulement de l'identification d'un client commence par une étape de recherche du client. Si le client est trouvé, le contrat d'échange retourne les renseignements client, sinon un contrat d'échange de "Création de client" est activé.


Les étapes du déroulement sont représentées par des *échanges utilisés*.

 Un échange utilisé représente l'utilisation d'un échange dans un contrat d'échange.

Exemple de contrat d'échange de communication avancé

Un contrat d'échange est décrit par un enchaînement d'étapes qui sont représentées soit :

- par des *échanges utilisés*
- par des *contrats d'échange utilisés*

 Un contrat d'échange utilisé est associée à un contrat d'échange. Il permet de représenter des échanges complexes.

Les rôles du contrat d'échange, présentés à la frontière du cadre, représentent les intervenants :

- client/fournisseur, ou
- émetteur/récepteur

Il est possible de décrire un contrat d'échange faisant intervenir plus de deux intervenants. Dans ce cas, un rôle est initiateur du contrat d'échange et les autres sont des contributeurs.

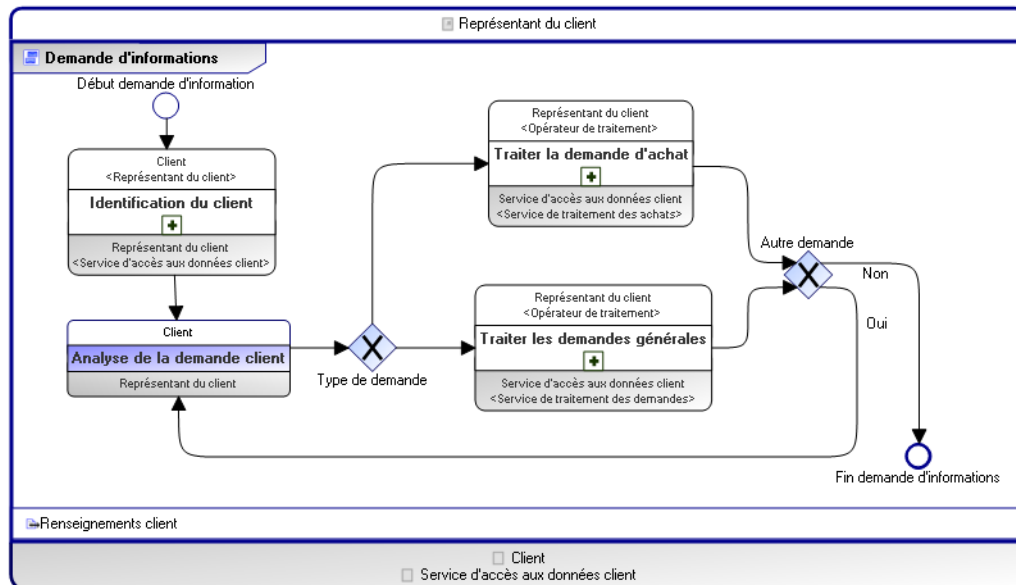


Diagramme de contrat d'échange (BPMN) "Demande d'informations"

Le contrat d'échange de "Demande d'information" est utilisé par le centre d'appels pour prendre en compte la demande d'un client en ligne. Il existe donc trois intervenants dans ce contrat d'échange : le client, les applications IT et le représentant du client qui est le demandeur effectif du service (il s'agit ici du centre d'appel).


Ce contrat d'échange consiste à identifier le client, puis à analyser sa demande. La demande est ensuite traitée comme demande d'achat ou comme une autre demande s'il s'agit d'une demande de renseignements, par exemple.

Créer un contrat d'échange à partir d'une interaction

Vous pouvez créer un nouveau contrat d'échange :

- à partir d'une bibliothèque,
- à partir d'une interaction dans un diagramme.

Pour créer un contrat d'échange, dans un diagramme, à partir d'une interaction :


1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Interaction**

2. Tirez un lien entre les deux entités en communication.
3. Dans la fenêtre d'ajout d'interaction, cliquez sur la flèche à droite du champ **Contrat d'échange** et sélectionnez **Nouveau**.
La fenêtre **Création d'un contrat d'échange** s'ouvre.
4. Saisissez le nom du contrat d'échange dans le champ **Nom**.
5. Cliquez sur le bouton **OK**.
L'interaction et le contrat d'échange sont créés.

Créer un diagramme de contrat d'échange (BPMN)


Un contrat d'échange est représenté par un **Diagramme de contrat d'échange (BPMN)**.

Pour créer un diagramme de contrat d'échange (BPMN) à partir d'une interaction :

1. Faites un clic droit sur l'interaction.
2. Sélectionnez le contrat d'échange associé et, dans son menu contextuel, cliquez sur **Nouveau > Diagramme de contrat d'échange (BPMN)**.
Le diagramme s'ouvre avec le cadre du contrat d'échange et les deux **rôles** qui représentent le consommateur et le fournisseur.

 *Un rôle est un participant à une interaction, un workflow ou un processus. Il peut en être l'initiateur, c'est-à-dire le demandeur du service, ou représenter un sous-traitant réalisant un traitement extérieur au service. Un rôle fait partie intégrante de l'objet qu'il décrit et n'est pas réutilisable. Il peut par la suite être affecté à un acteur interne ou externe à l'organisation ou à un composant informatique. Exemples : Demandeur, voyageur.*


Les **événements**, les **branchements** et les **enchaînements** de votre diagramme respectent les standards de la norme BPMN.


 *Pour plus de détails sur les événements, branchements et enchaînements, voir "Gérer les événements, les branchements et les enchaînements", page 171.*

Définir un échange ou un contrat d'échange utilisé


Dans un diagramme de contrat d'échange (BPMN), les opérations sont décrites par :

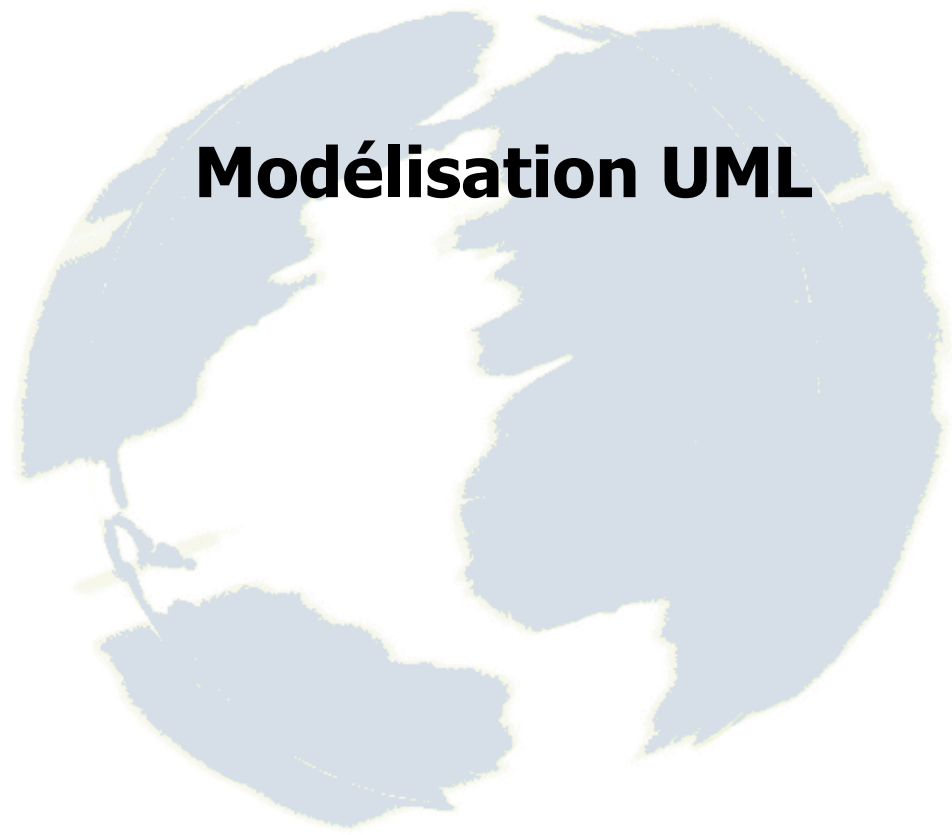
- des **contrats d'échange utilisés**
- des **échanges utilisés**

 *Un contrat d'échange utilisé est associée à un contrat d'échange. Il permet de représenter des échanges complexes.*

 *Un échange utilisé représente l'utilisation d'un échange dans un contrat d'échange.*

Pour créer un *contrat d'échange utilisé* :

1. Sélectionnez le bouton **Contrat d'échange utilisé**  et cliquez dans le diagramme à l'intérieur de la forme du contrat d'échange. La fenêtre création s'ouvre.
2. Cliquez sur la flèche à droite du champ **Spécification du contrat d'échange utilisé**.
3. Sélectionnez **Relier contrat d'échange** dans la liste déroulante et choisissez le contrat d'échange que vous souhaitez utiliser.
4. Sélectionnez dans le champ **A partir de** le rôle du contrat d'échange décrit relié au rôle "Consommateur" du contrat d'échange utilisé.
5. Sélectionnez dans le champ **Jusqu'à** le rôle du contrat d'échange décrit relié au rôle "Fournisseur" du contrat d'échange utilisé.
6. Cliquez sur **Terminer**.



A PROPOS DE L'IMPLÉMENTATION UML



Le langage de modélisation UML (Unified Modeling Language) s'est érigé comme standard pour la modélisation graphique des systèmes d'information. **HOPEX Application Design** offre un ensemble d'outils vous qui vous permettent de modéliser votre SI conformément à la version 2.3 de ce standard.

Ce guide a pour objectif de vous faire découvrir les principales fonctionnalités d'HOPEX UML.

- ✓ ["Présentation", page 132](#)
- ✓ ["Organisation des diagrammes d'UML", page 134](#)

PRÉSENTATION

HOPEX Application Design vous permet de :

Analyser les cas d'utilisation

Une réflexion sur les fonctionnalités attendues du futur système est nécessaire avant sa conception. Les composants de ce système vont en effet être utilisés par les acteurs de l'organisation pour effectuer leur mission. Les divers "cas d'utilisation" de ce système vont être présentés dans les **diagrammes de cas d'utilisation**.

Ils serviront de point de départ pour la découverte des objets.

Ils permettront ensuite de valider l'utilisation de ces objets à l'aide des diagrammes d'interaction.

Ils donneront enfin des critères de regroupement en "paquetage" pour les objets découverts.

Voir ["Le diagramme de cas d'utilisation", page 137](#).

Identifier les objets

Les objets ayant une structure semblable, le même comportement et les mêmes types de relations avec d'autres objets sont réunis dans une classe.

Le **diagramme de classes** permet d'identifier les objets mis en jeu à l'intérieur du système et de définir leur structure en termes d'attributs et d'opérations ainsi que les relations entre eux. Le **diagramme d'objets** montre les instances compatibles avec un diagramme de classes particulier qu'on peut ainsi valider avec un exemple.

Voir ["Le diagramme de classes", page 151](#).

Décrire les comportements

Le **diagramme de machine à états** permet de définir le comportement d'un objet vis-à-vis des sollicitations internes ou externes auxquelles il peut être soumis. Chacun des états dans lequel peut se trouver l'objet est indiqué ainsi que les réactions de l'objet à un événement donné en fonction de l'état où il se trouve.

Le **diagrammes d'activité** décrit également un comportement, mais en termes d'actions.

Voir :

- ["Le diagramme de machine à états", page 217](#)
- ["Le diagramme d'activités", page 227](#)

Représenter les interactions entre les objets

Le dialogue qui s'instaure entre les différents objets concernés par l'événement pour y répondre peut être représenté dans des **diagrammes d'interaction**.

Les diagrammes d'interaction insistent sur les échanges qui ont lieu entre les objets.

Le **diagramme de séquence** présente ces mêmes échanges en mettant en évidence leur chronologie.

Le **diagramme de communication** met l'accent sur l'organisation structurelle des objets qui envoient et reçoivent des messages.

Le **diagramme de vue générale d'interaction** donne une vue d'ensemble du flot de contrôle.

Voir ["Les diagrammes d'interaction"](#), page 235.

Répartir les classes entre les paquetages

Une fois les objets identifiés, il est aisé de répartir les classes qui les implémentent entre les différents paquetages. Les regroupements de ces classes effectués dans le **diagramme de paquetages** sont faits de manière à minimiser les échanges entre les différents paquetages. Ils obéissent à deux critères : l'un, technique, lié à leur environnement d'exécution et l'autre, organisationnel, lié à l'emploi qui en sera fait par les différents utilisateurs à équiper pour chaque cas d'utilisation.

Voir ["Le diagramme de paquetages"](#), page 204.

Définir les interfaces

Pour respecter le principe d'encapsulation, la répartition des éléments entre les composants est stricte. Il est donc nécessaire de prévoir les interfaces entre les éléments ayant des relations entre eux et qui appartiennent cependant à des composants différents.

Le **diagramme de composants** et le **diagramme de structure composite** présentent l'interdépendance des composants ou éléments d'un composant.

La définition des interfaces des objets ainsi que l'adhésion à un protocole d'échange normalisé (CORBA2, DCOM/OLE) sont des facteurs clés de l'interopérabilité, c'est-à-dire la capacité à faire coopérer les objets développés et exploités dans des environnements hétérogènes.

Voir :

- ["Le diagramme de composants"](#), page 207
- ["Le diagramme de structure composite"](#), page 212

Spécifier la mise en oeuvre

La mise en oeuvre des objets dans un environnement de travail concret peut être spécifiée les **diagrammes de déploiement**.

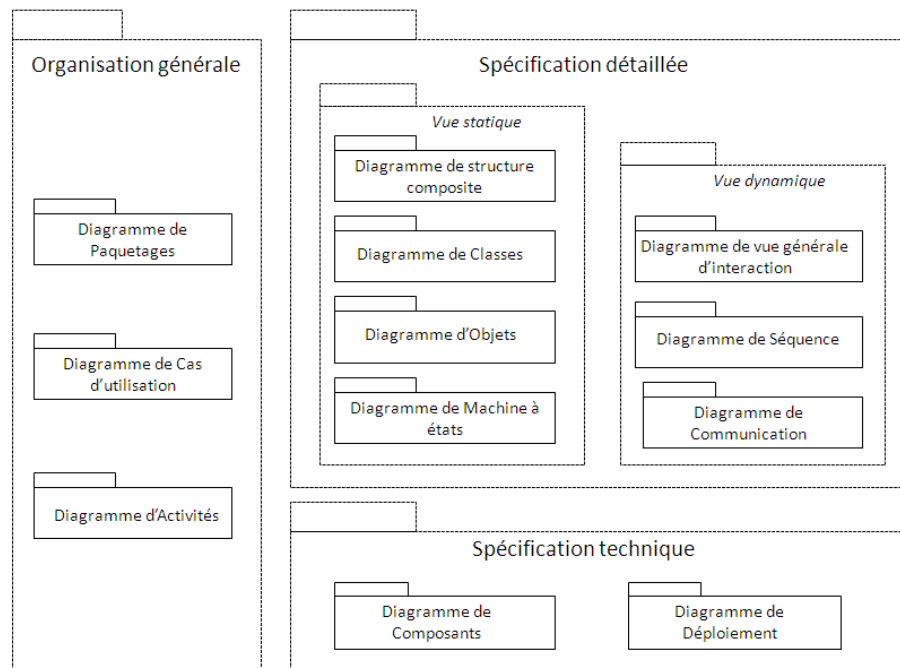
Voir ["Le diagramme de déploiement"](#), page 251.

ORGANISATION DES DIAGRAMMES D'UML

Organisation générale

Les **diagrammes de cas d'utilisation** vous permettent de montrer les principales interactions entre le système étudié et son environnement, et de mettre en évidence ses principaux sous-systèmes.

Les **diagrammes de paquetages** constituent un découpage du système. Le découpage du système en paquetages est relativement structurant dans la mesure où un objet ne peut être que dans un seul paquetage. Vous pouvez commencer à dessiner des diagrammes de paquetages dès que vous avez identifié les principaux composants de votre système (Commercial, Production, Facturation, ...).



Spécification détaillée

Le diagramme principal est le **diagramme de classes**. Il décrit l'essentiel de la sémantique des objets composant le système. C'est le diagramme sur lequel les concepteurs passeront généralement le plus de temps. La découverte des classes se fait généralement par des aller-retour entre les diagrammes de classes et les diagrammes de séquence.

Le **diagramme de machine à états** permet de décrire l'aspect statique d'un objet, c'est à dire les différents états dans lesquels il peut se trouver et les transitions possibles entre ces états. Il permet ainsi de compléter la description d'une classe.

Les **diagrammes d'interaction** permettent de spécifier l'aspect dynamique du système en montrant l'interaction des objets entre eux. Ils permettent en particulier de décrire de façon détaillée les différents scénarios de fonctionnement d'un cas d'utilisation. Le diagramme de séquence explicite plutôt le déroulement d'un scénario dans le temps, tandis que le diagramme de communication insiste plutôt sur l'interaction entre les objets.

Spécification technique et mise en oeuvre

Le **diagramme de composants** décrit les différents composants techniques d'une application et leurs interactions.

Le **diagramme de structure composite** précise les collaborations entre les composants ou éléments d'un composant dans l'exécution d'une tâche commune.

Le **diagramme de déploiement** permet de préciser l'architecture technique du système en indiquant sur quels postes de travail ou sur quels nœuds du système informatique seront installés les différents composants de l'application.

Points d'entrée des diagrammes d'UML

Diagramme	Points d'entrée
Diagramme de classes	Paquetage, Classe, Cas d'utilisation
Diagramme d'objets	Classe, Composant, Paquetage, Cas d'utilisation
Diagramme de composants	Composant, Paquetage
Diagramme de structure composite	Composant, Classe, Collaboration
Diagramme de déploiement	Paquetage
Diagramme de paquetages	Paquetage, Bibliothèque
Diagramme de cas d'utilisation	Paquetage, Cas d'utilisation, Environnement d'application (ADES)
Diagramme de séquence (UML2)	Interaction
Diagramme de communication	Interaction
Diagramme de vue générale d'interaction	Interaction
Diagramme d'activités (UML2)	Activité
Diagramme de machine à états	Machine à état, Machine à état de protocole

Dans **HOPEX Application Design**, les différents points d'entrée ci-dessus sont accessibles dans le volet de navigation **Ressources des applications > Implémentation UML**.

LE DIAGRAMME DE CAS D'UTILISATION



Le diagramme de cas d'utilisation constitue une première étape dans la description d'un système d'information. Il permet d'identifier les fonctionnalités que doit fournir le système afin de répondre aux besoins des acteurs de l'organisation ; il décrit en ce sens les interactions entre le système et les acteurs.

Les points suivants sont abordés ici :

- ✓ ["Créer un diagramme de cas d'utilisation", page 138](#)
- ✓ ["Les éléments du diagramme de cas d'utilisation", page 140](#)

CRÉER UN DIAGRAMME DE CAS D'UTILISATION

Un diagramme de cas d'utilisation permet de décrire les interactions entre les acteurs de l'organisation et le système dans chacun des *cas d'utilisation* envisagés.



Un cas d'utilisation est une suite d'actions qui amène un résultat observable pour un acteur particulier. Des scénarios illustrent les cas d'utilisation par l'exemple.

Ces cas d'utilisation sont regroupés dans des *paquetages* représentant les frontières du système.



Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.

Vous pouvez créer un diagramme de cas d'utilisation à partir d'un paquetage. Cependant, dans le cadre de systèmes complexes, vous pouvez créer ce type de diagramme depuis un cas d'utilisation afin de détailler ce dernier.

Dans **HOPEX Application Design** vous pouvez également créer un diagramme de cas d'utilisation sur l'environnement d'application d'un projet. Voir "[Créer un diagramme de cas d'utilisation sur un environnement d'application](#)".

Créer un paquetage

Condition préalable

Avoir défini un environnement de travail sur un projet en cours. Voir "[Le volet Projet en cours](#)", page 22.

Pour créer un paquetage dans **HOPEX Application Design** :

1. Cliquez sur le menu de navigation puis sur **Projets en cours**.
2. Dans le volet de navigation cliquez sur **Périmètre du projet**.
3. Dans la zone d'édition, cliquez sur **Paquetages**.
4. Dans la zone d'édition, cliquez sur **Nouveau**.
La fenêtre de création d'un paquetage s'ouvre.
5. Saisissez son **Nom**.
6. Indiquez éventuellement la bibliothèque ou le paquetage détenteur.



La bibliothèque par défaut est utilisée pour ranger un objet s'il n'y a pas de bibliothèque courante au moment de sa création.

7. Cliquez sur **OK**.

Le paquetage est créé et ajouté dans la liste des paquetages.

Créer le diagramme de cas d'utilisation du paquetage

Pour créer un diagramme de cas d'utilisation :


1. Cliquez sur l'icône du paquetage concerné et sélectionnez **Nouveau > Diagramme de cas d'utilisation.**

Le diagramme créé s'ouvre dans la fenêtre d'édition. Le cadre du paquetage est positionné à l'intérieur du dessin.

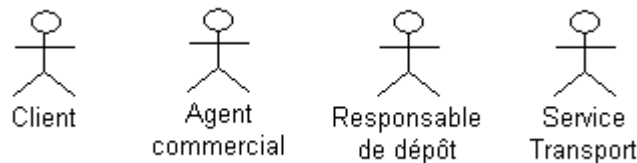
LES ÉLÉMENTS DU DIAGRAMME DE CAS D'UTILISATION

- ✓ "Les acteurs", page 140
- ✓ "Les cas d'utilisation", page 141
- ✓ "Les paquetages", page 141
- ✓ "Les participations", page 143
- ✓ "Les associations entre cas d'utilisation : extension et inclusion", page 144
- ✓ "Les généralisations", page 147
- ✓ "Les interfaces", page 148


Les acteurs

 Un acteur représente le rôle joué par quelque chose ou quelqu'un se trouvant dans l'environnement de l'entreprise ou du système étudié. Il est en relation avec le métier de l'entreprise et interagit avec le système dans différents cas d'utilisation. Ce peut être un élément de la structure de l'entreprise tel qu'une direction, un service ou un poste de travail.

Exemples d'acteurs :



Pour créer un **acteur** dans un diagramme de cas d'utilisation :

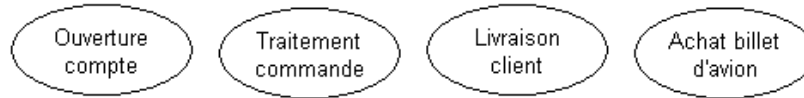
1. Dans la barre d'objets, cliquez sur le bouton **Acteur** .
2. Cliquez sur le plan de travail du diagramme.
La fenêtre **Ajout d'un acteur** s'ouvre.
3. Saisissez le nom de l'acteur, "Standardiste", par exemple.
4. Cliquez sur le bouton **Ajouter**.
L'acteur apparaît alors dans le diagramme.

☺ Vous pouvez créer plusieurs éléments à la suite sans revenir à la barre d'outils en effectuant un double-clic sur le bouton **Acteur**


☺ Pour revenir ensuite au mode normal, utiliser la touche <Echap>, ou cliquez sur un autre bouton de la barre d'outils,

Les cas d'utilisation

Exemples de **cas d'utilisation** : traitement d'une commande, livraison d'un client, ouverture d'un compte, envoi d'une facture, établissement d'un crédit, achat d'un billet d'avion, etc.



Pour créer un cas d'utilisation dans un diagramme :

1. Dans la barre d'objets du diagramme de cas d'utilisation cliquez sur le bouton **Cas d'utilisation** .
La fenêtre **Ajout d'un cas d'utilisation** s'ouvre.
2. Saisissez le **Nom** du cas d'utilisation et cliquez sur le bouton **Ajouter**.

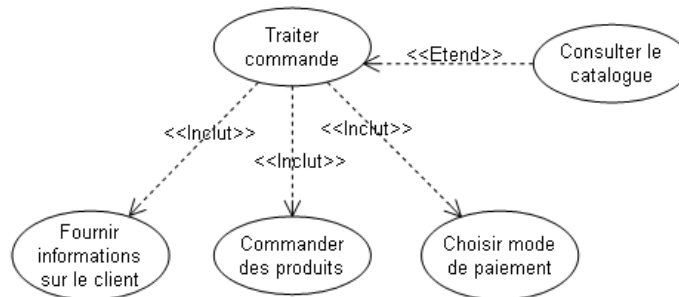
Le cas d'utilisation apparaît sur le diagramme.

Faire un zoom sur un cas d'utilisation

Pour ouvrir directement, depuis le diagramme du paquetage, le diagramme qui décrit un cas d'utilisation :


1. Faites un clic droit sur le cas d'utilisation.
2. Sélectionnez **Diagramme de cas d'utilisation**.

Le diagramme de cas d'utilisation s'ouvre.



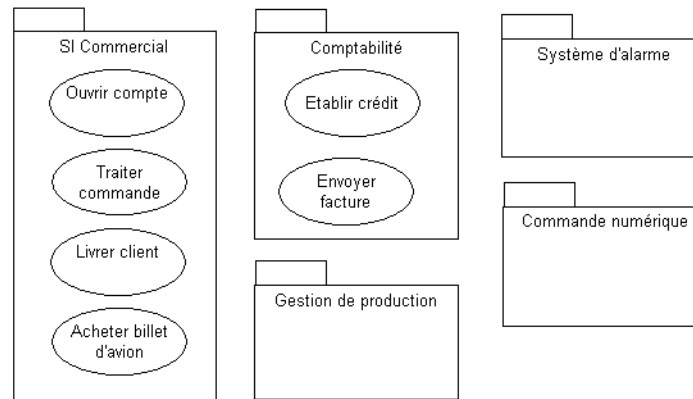
☺ Ce zoom sur la description d'un objet est disponible sur tous les éléments décrits par un diagramme.


Les paquetages

 Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas

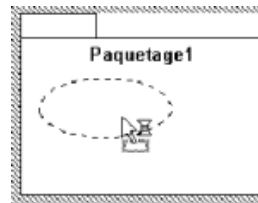
d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.

Exemples de **paquetage** : Le système d'information commercial, la comptabilité, la gestion de production, la commande numérique d'une machine, un système d'alarme, etc.



Vous pouvez créer un paquetage à l'aide du bouton **Paquetage**  de la barre d'outils. Vous pouvez l'agrandir pour pouvoir poser des cas d'utilisation dessus.

Vous pouvez relier un cas d'utilisation à un paquetage simplement en le posant dessus. Lorsque vous déplacez l'objet sur le paquetage, la forme du paquetage est mise en relief pour indiquer que l'objet va bien lui être relié.



☛ Si les objets reliés disparaissent sous le paquetage, cliquez sur le paquetage puis cliquez sur le bouton **Arrière-plan** de la barre de dessin.

Lorsque vous déplacez un cas d'utilisation d'un paquetage à l'autre avec la souris, il est délié du premier et relié au deuxième. En revanche, si vous le déplacez avec les flèches du clavier, les liens ne sont pas modifiés.

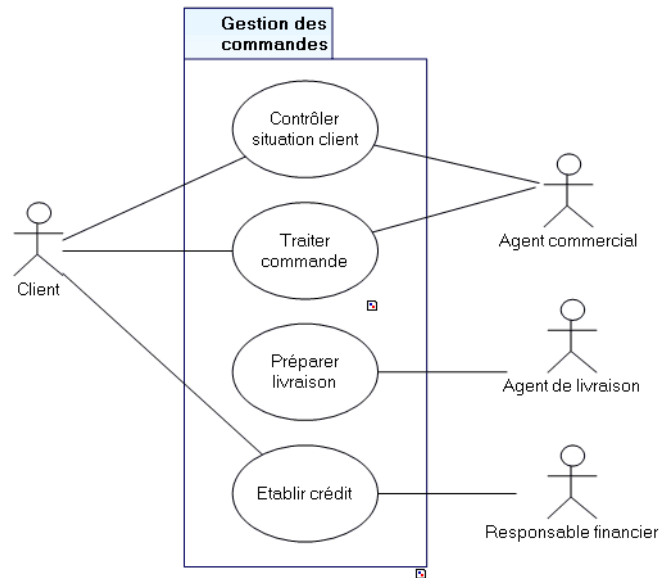
Les participations

Vous pouvez indiquer quel acteur participe à chacun des différents cas d'utilisation.



Une participation indique qu'un acteur joue un rôle dans un cas d'utilisation.


Exemple de participation




- L'agent commercial intervient dans le traitement d'une commande et dans le contrôle de la situation du client ;
- L'agent de livraison intervient dans la livraison ;
- Le responsable financier intervient dans l'établissement des crédits, etc.

Créer une participation

Pour créer une **participation** dans le diagramme de cas d'utilisation :



1. Cliquez sur le bouton **Participation**  de la barre d'objets.
2. Cliquez sur l'acteur concerné, et faites glisser la souris jusqu'au cas d'utilisation, avant de relâcher votre pression. Une fenêtre apparaît.
3. Indiquez le nom de la participation et indiquez si l'acteur en est l'initiateur.


 Il est possible de préciser le début du cas d'utilisation en activant la case à cocher **Initiateur** dans la fenêtre de propriétés de la participation correspondante. Une flèche apparaît sur le trait représentant la participation.

4. Cliquez sur **OK**.

Le lien représentant la participation apparaît dans le diagramme.

 **Une participation est représentée par un lien mais il s'agit d'un objet, avec des propriétés qui lui sont propres.**

 La bobine  n'est pas utilisée pour créer des participations. Elle sert à créer certains types de liens comme ceux entre Paquetage et les autres objets.

 En cas d'erreur, vous pouvez supprimer un objet en cliquant avec le bouton droit sur cet objet, et en sélectionnant la commande **Supprimer** dans le menu contextuel. Vous pouvez de même supprimer un lien en cliquant avec le bouton droit sur le lien, et en sélectionnant la commande **Supprimer** ou **Délier** dans le menu contextuel du lien.

Multiplicités d'une participation

Sur une participation, il est possible de préciser la multiplicité :

- De l'acteur, afin d'indiquer que plusieurs instance de l'acteur sont impliquées dans une même instance du cas d'utilisation (exemple : les participants à une réunion).
- Du cas d'utilisation, afin d'indiquer qu'une même instance d'acteur intervient dans plusieurs instances du cas d'utilisation (exemple : un agent commercial qui traite plusieurs commandes d'un même client).

Pour définir les multiplicités sur une participation :

1. Sélectionnez la participation et affichez ses **Propriétés**.
2. Dans la page de propriétés, cliquez sur la liste déroulante puis sur **Caractéristiques**.

Une première section vous permet de définir la multiplicité de l'acteur, une seconde section permet de définir celle du cas d'utilisation.

Une fois définie, la multiplicité apparaît dans le diagramme.

Les associations entre cas d'utilisation : extension et inclusion

Lorsque la taille du système à décrire est importante, il est utile d'avoir des mécanismes de représentation adaptables au niveau de détail désiré. C'est ce que permettent les associations entre cas d'*utilisation*.

Lorsqu'un cas d'utilisation comprend trop de possibilités et d'exceptions, ces dernières sont représentées à part dans des extensions du cas d'utilisation standard.

Relation d'inclusion

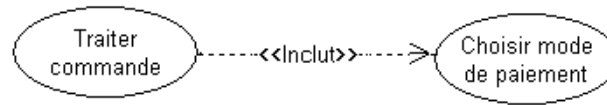
Un cas d'utilisation peut être sollicité automatiquement à la suite d'un autre, par exemple, la validation d'une commande inclut obligatoirement la sélection d'un moyen de paiement.

Pour indiquer qu'un cas d'utilisation en inclut un autre :

1. Dans le diagramme de cas d'utilisation, cliquez sur le bouton **Lien** 

2. Cliquez sur le cas utilisateur, par exemple "Traiter commande" et faites glisser la souris jusqu'au cas utilisé, par exemple "Choisir mode de paiement", avant de relâcher votre pression.
3. Sélectionnez le type de lien "Cas inclus" et cliquez sur **OK**.

Le lien apparaît dans le dessin accompagné du mot "Inclut".



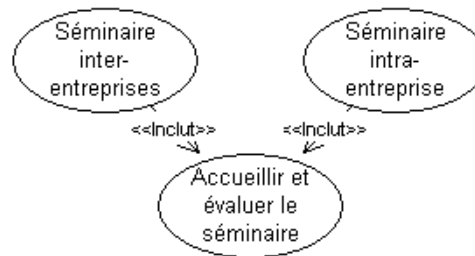
Exemples d'inclusion

Dans un organisme de formation, les cas d'utilisation :

- Séminaire inter-entreprises (dont les participants viennent de plusieurs entreprises différentes)
- Séminaire intra-entreprise (dont les participants viennent tous de la même entreprise)

peuvent avoir en commun le cas d'utilisation :

- Accueillir et évaluer le séminaire



Dans une entreprise commerciale, le cas d'utilisation :

- Passer une commande

peut réutiliser les cas d'utilisation suivants :

- Fournir des informations sur le client
- Passer un ordre de production
- Proposer un mode de paiement

Relation d'extension

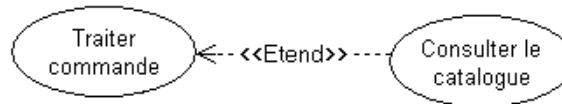
Un cas d'utilisation peut entraîner l'exécution d'un autre. Contrairement à l'inclusion qui est automatique, l'extension est optionnelle.

Pour indiquer qu'un cas d'utilisation est l'extension d'un autre :

1. Dans le diagramme de cas d'utilisation, cliquez sur la flèche associée au bouton **Lien** et cliquez sur **Extension**.

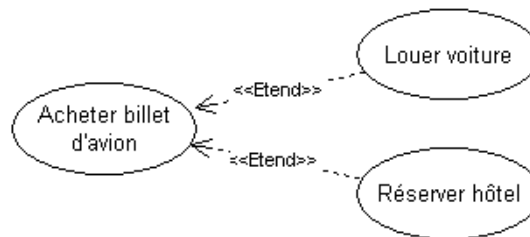
2. Cliquez dans un cas d'utilisation, par exemple "Consulter le catalogue" et faites glisser la souris jusqu'au cas étendu, par exemple "Traiter commande" avant de relâcher votre pression.
La fenêtre de création d'une extension apparaît. Vous pouvez définir une contrainte ou un point d'extension.
3. Cliquez sur **OK**.

Le lien apparaît dans le diagramme accompagné du mot "Etend".



Exemple d'extension

L'achat d'un billet d'avion peut être complété par la réservation d'un hôtel ou la location d'une voiture.



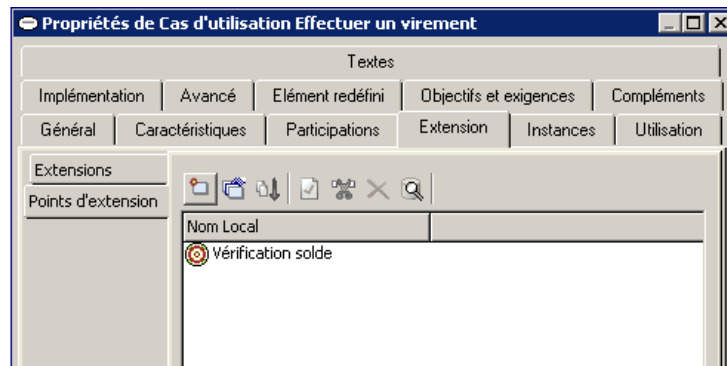
Point d'extension

L'extension peut intervenir à un point précis du cas étendu. Ce point est appelé point d'extension.

Pour créer un point d'extension sur le cas étendu :

1. Affichez la fenêtre de propriétés de l'extension.
2. Sélectionnez la page **Caractéristiques**.
3. Sous la section **Point Extension**, cliquez sur **Ajouter**.
La fenêtre de recherche apparaît.

4. Sélectionnez le point d'extension voulu et cliquez sur **Relier**.
Le point d'extension apparaît dans la fenêtre de propriétés de l'extension.

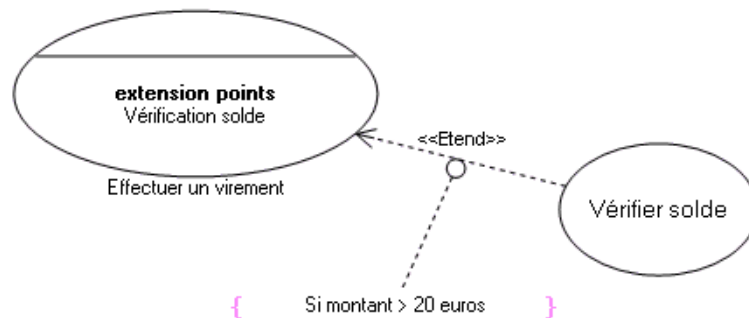


Un point d'extension peut être associé à une **contrainte** qui indique le moment où l'extension intervient. Vous pouvez ajouter une contrainte lors de la création de l'extension, ou ultérieurement, dans la fenêtre de propriétés du lien d'extension.

Une contrainte est une déclaration qui établit un contrôle ou une règle de gestion impliquant généralement plusieurs classes.

Exemple de point d'extension

L'exemple suivant présente le cas d'utilisation d'un virement bancaire ; au delà de la somme de 20 euros, la vérification de solvabilité du client est déclenchée.

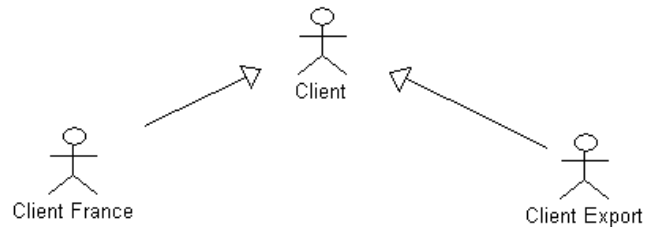


Les généralisations

Une généralisation représente une relation d'héritage entre une classe générale et une classe plus spécifique. La classe spécifique est cohérente avec la classe plus générale et en hérite ses caractéristiques et son comportement. Elle comporte cependant des informations supplémentaires. Toute instance de la classe spécifique est aussi une instance de la classe générale.


La notion de *généralisation*, initialement utilisée pour les classes, a été étendue à d'autres concepts d'UML comme acteur et cas d'utilisation.

Exemples de généralisation entre acteurs :

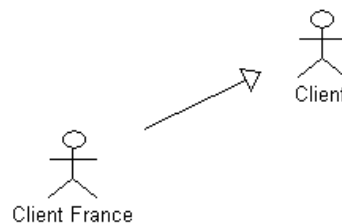


L'acteur "Client" peut être spécialisé en France et à l'Export.

Pour créer une généralisation entre acteurs dans un diagramme de cas d'utilisation :

- 1 Cliquez sur le bouton  et tirez le lien en partant de l'acteur particulier (Ex : Client France) vers l'acteur général (Ex : Client).

La généralisation apparaît dans le dessin.



Vous pouvez créer de la même manière une généralisation entre deux cas d'utilisation.

Lors de la création d'une deuxième généralisation, une fenêtre vous propose de réutiliser la généralisation existante si elle porte sur le même sujet.

Les interfaces

Il est possible de compléter la description d'un cas d'utilisation ou d'un acteur par la description des *interfaces* par lesquelles il communique avec son environnement.

Créer une interface

Pour créer une interface dans un diagramme de cas d'utilisation :

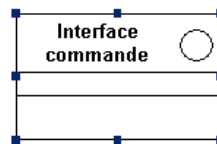
1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Interface**



☛ Si le bouton **Interface** n'apparaît pas dans la barre d'objets, cliquez sur **Affichage > Vues et détails** et dans la fenêtre qui s'affiche, cochez la case **Classes**.

2. Cliquez sur le plan de travail du diagramme.
3. Dans la fenêtre qui s'ouvre, saisissez le nom de l'interface et cliquez sur le bouton **Ajouter**.


L'interface apparaît sur le diagramme.



Relier une interface à un cas d'utilisation

Lorsque vous reliez une interface à un cas d'utilisation, vous devez préciser s'il s'agit d'une interface supportée ou d'une interface requise par le cas d'utilisation.

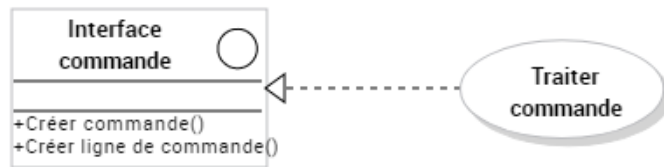
Pour préciser le type de lien entre une interface et un cas d'utilisation :

1. Cliquez sur le bouton **Lien**  et tirez le lien en partant du cas d'utilisation (Ex : Traiter Commande) vers l'interface (Ex : Interface Commande). Une fenêtre apparaît.
2. Sélectionnez le type du lien à créer :
 - Interface requise
 - Interface supportée
3. Cliquez sur **OK**.

Le lien apparaît dans le dessin.



Vous pouvez préciser quelles sont les opérations que ce cas d'utilisation peut réaliser par l'intermédiaire de cette interface.



- ✓ Pour plus de détails sur les interfaces requises et supportées, voir aussi ["Relier les interfaces aux autres objets"](#), page 208.

LE DIAGRAMME DE CLASSES



Un diagramme de classes permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et leurs relations. Un diagramme d'objets montre des exemples illustrant ce diagramme de classes.

La spécification des diagrammes de classes est souvent considérée comme la partie la plus importante dans la modélisation d'un système d'information. Les points abordés dans ce chapitre sont :

- ✓ "Présentation du diagramme de classes", page 152
- ✓ "Créer un diagramme de classes", page 153
- ✓ "Les classes", page 154
- ✓ "Les attributs", page 160
- ✓ "Les opérations", page 163
- ✓ "Les signaux", page 169
- ✓ "Les associations", page 171
- ✓ "Les généralisations", page 187
- ✓ "Spécifier les interfaces", page 192
- ✓ "Spécifier les dépendances", page 194
- ✓ "Spécifier des classes paramétrées", page 195
- ✓ "Les contraintes", page 196
- ✓ "Le diagramme d'objets", page 197

PRÉSENTATION DU DIAGRAMME DE CLASSES

Un diagramme de classes permet de représenter la structure statique d'un système, en particulier les types d'*objets* manipulés dans le système, leur structure interne et leurs relations.



Un objet est une entité avec une identité et des frontières clairement définies dont l'état et le comportement sont encapsulés. Son état est défini par les valeurs de ses attributs et de ses liens avec d'autres objets. Son comportement est représenté par ses opérations et ses méthodes. Un objet est une instance de classe.

Exemples d'objets :

- Objets de gestion :
 - Jacques Dupond, Pierre Durand, Paul Smith sont des instances de la classe "personne".
 - Les commandes no 10533 et 7322 sont des instances de la classe "commande".
 - Ecran SPD-1730 est une instance de la classe "article".
- Objets techniques utilisés pour la programmation :
 - Dlg_Order_Create, Dlg_Customer_Query sont des instances de la classe "fenêtre".
 - Str_Customer_Name, Str_Product_Comment sont des instances de la classe "chaîne".

Modéliser les données consiste à identifier les classes considérées d'intérêt pour représenter l'activité de l'entreprise, et à définir les associations qui existent entre elles.

Il faut que les classes et les associations qui constituent le diagramme de classes associé à un secteur de l'entreprise suffisent à le décrire complètement sur le plan sémantique.

En d'autres termes, on doit pouvoir décrire l'activité de l'entreprise en utilisant seulement ces classes et associations.

Ceci n'implique pas que, pour chaque mot ou verbe utilisé pour cette explication, il y ait un correspondant direct dans le diagramme de classes. Il s'agit de pouvoir traduire ce que l'on veut exprimer, au travers des classes et des associations.

La spécification des diagrammes de classes est souvent considérée comme la partie la plus importante dans la modélisation d'un système d'information.

Un diagramme d'objets montre des exemples illustrant ce diagramme de classes.

En particulier, il est possible d'illustrer un diagramme de classes par le diagramme d'objets correspondant dans un même dessin.

Le diagramme de classes : synthèse

Un diagramme de classes inclut :

- Des classes, qui représentent les concepts de base (client, compte, produit, etc.).
- Des associations, qui définissent les relations entre les différentes classes.
- Des attributs, qui décrivent les caractéristiques des classes et, dans certains cas, des associations.
- Des opérations, qui peuvent être effectuées sur les objets de la classe.

☛ *Les opérations ne sont pas prises en compte par les outils de **HOPEX Information Architecture** (synchronisation, génération etc.).*

Le diagramme de classes est complété avec la définition des multiplicités.

A la fin de ce guide, un glossaire rappelle la définition de chacun des concepts utilisés.

Créer un diagramme de classes

Un diagramme de classe se crée à partir d'un paquetage.

Pour créer un diagramme de classes :

- 】 Cliquez sur l'icône du paquetage concerné et sélectionnez **Nouveau > Diagramme de classes.**

Le diagramme créé s'ouvre dans la fenêtre d'édition.

Un diagramme de classes peut décrire un paquetage, un cas d'utilisation, une classe ou une instance.

LES CLASSES

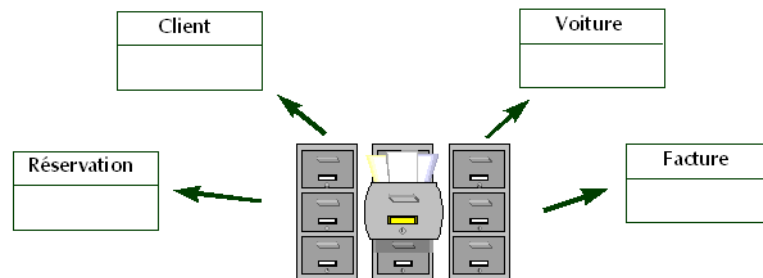
- ✓ "Définition d'une classe", page 154
- ✓ "Créer une classe", page 155
- ✓ "Propriétés d'une classe", page 156
- ✓ "Stéréotype d'une classe", page 157

Définition d'une classe

Une *classe* est décrite par une liste d'attributs et d'opérations.

Une classe est reliée à d'autres classes via des *associations*. L'ensemble des classes et des associations constitue le noyau d'un diagramme de classes.

Nous pouvons illustrer la notion de classe en comparant les classes à des fiches dans des tiroirs.



Les classes peuvent être des objets techniques utilisés pour la programmation.

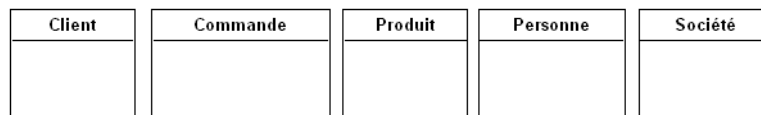
Exemples : fenêtre, rectangle, chaîne, tableau, etc.

Les classes peuvent représenter des objets techniques utilisés dans l'industrie.

Exemples : Alarme, Capteur, Zone

Les classes peuvent également représenter des objets de gestion.

Exemples : client, commande, produit, personne, société, etc.




Une classe peut aussi être la traduction d'un processus comme "Valider demande client" ou l'implémentation d'une règle de gestion fonctionnelle comme "Cohérence comptes analytiques".

☛ À la fin de ce guide, un glossaire rappelle la définition de chacun des concepts utilisés.


Créer une classe

Pour créer une classe :

1. Dans le diagramme de classes, cliquez sur le bouton **Classe**  de la barre d'insertion d'objets.
2. Cliquez sur le plan de travail du diagramme.
La fenêtre **Ajout d'une classe** s'ouvre.
3. Saisissez le **Nom** de la classe et cliquez sur le bouton **Ajouter**.
La classe est alors posée dans le diagramme.

☛ Dans les exemples présentés dans ce guide, les noms des objets incluent, entre autres, des blancs, des majuscules et des caractères accentués. Il est important de noter que, lorsqu'il y a, pour un outil de génération qui utilise les spécifications effectuées avec **HOPEX UML**, des restrictions sur les caractères autorisés et sur la longueur des noms, il est préférable de se conformer à ces règles dès la spécification avec **HOPEX UML**.

Vous pouvez créer plusieurs classes à la suite sans revenir à la barre d'outils en double-cliquant sur le bouton **Classe**.

Pour revenir ensuite au mode normal, cliquez sur la flèche .

Vous pouvez utiliser partout le nom complet d'une classe en indiquant le paquetage auquel elle appartient séparé par deux fois deux points.

Exemple :

Entreprise::Gestion Commerciale::Client

Si l'un des paquetages cités n'existe pas, il est automatiquement créé et relié à la classe.

Retrouver une classe existante

Pour retrouver une classe existante:

1. Dans la fenêtre **Ajout d'une classe**, sélectionnez la commande **Lister** à l'aide de la flèche.
La liste des classes apparaît.
2. Sélectionnez la classe qui vous intéresse et cliquez sur **OK**.
Le nom de la classe sélectionnée apparaît dans la fenêtre d'ajout d'une classe.
3. Cliquez sur **Ajouter**.
La classe apparaît dans le diagramme.

Propriétés d'une classe

Les propriétés affichées varient en fonction du stéréotype de la classe.

Pour ouvrir la fenêtre de propriétés d'une classe :

- 1 Sélectionnez la classe et cliquez sur le bouton **Propriétés** associé à la fenêtre d'édition.
Plusieurs pages permettent de définir les propriétés d'une classe.

Page Caractéristiques

La page **Caractéristiques** permet de saisir différentes caractéristiques de la classe :

- Son **Nom**, que vous pouvez modifier.

☺ Vous pouvez également modifier le nom d'une classe en cliquant directement sur son nom dans le dessin.

- Le détenteur de la classe (par exemple le paquetage).
- La **Visibilité** de la classe (par rapport au paquetage) :
 - "Publique" : la classe est visible par tout élément situé à l'extérieur du paquetage. C'est la visibilité sélectionnée par défaut.
 - "Protégée" : la classe est visible par les éléments héritiers ou importateurs et amis.
 - "Privée" : la classe est seulement visible par les éléments importateurs et amis.
 - "Non spécifiée".

📖 Les amis d'une classe sont des classes autorisées à connaître les internes de cette classe. Il est possible de préciser les amis d'une classe dans l'onglet Compléments de la fenêtre de propriétés de la classe.

- Son **Stéréotype** : voir "[Stéréotype d'une classe](#)", page 157.
- Un **Commentaire** : les commentaires permettent de compléter le diagramme lorsque des détails utiles n'apparaissent pas dans le dessin. Ces commentaires sont repris dans le document qui décrit le diagramme de classes.

Les autres caractéristiques que vous pouvez préciser sont l'abstraction, la persistance et l'activité :

- Une classe **Abstraite** n'a pas d'instance. Elle n'est utilisée que pour mettre en commun des opérations ou des attributs communs à ses sous-classes.
- La **Persistance** précise si cette classe doit être conservée dans le temps ou si elle ne vit que le temps du traitement en mémoire dans l'ordinateur.
- Les instances d'une classe **Active** sont capables de déclencher des flux de contrôle sans qu'il y ait d'intervention de l'utilisateur.

Ex : Une instance de la classe imprimante peut envoyer un message "Plus de papier disponible" sur l'écran de l'administrateur du réseau.

- Une classe **Racine** est une classe ne possédant pas de sur-classe dans l'arborescence des généralisations de classe.
- Une classe **Feuille** est une classe ne possédant pas de sous-classe dans l'arborescence des généralisations de classe.

Vous pouvez également préciser les Paramètres d'une classe paramétrée (pour C++).

☛ Voir "[Pour spécifier une classe paramétrée :](#)", page 195 pour plus de détails.

Autres pages de propriétés

D'autres pages vous permettent de définir ou de visualiser :

- Les attributs d'une classe (voir "[Les attributs](#)", page 160)
- Les opérations d'une classe (voir "[Les opérations](#)", page 163)
- Les classes associées (voir "[Les associations](#)", page 171)
- Les instances d'une classe (voir "[Le diagramme d'objets](#)", page 197)
- Les éléments redéfinis

Stéréotype d'une classe

Un stéréotype est un type d'élément de modélisation qui permet d'étendre la sémantique du métamodèle. Les stéréotypes doivent être basés sur des types ou des classes existantes dont ils reprennent la structure. D'autres stéréotypes peuvent être créés par l'utilisateur.

Les stéréotypes disponibles pour une classe sont :

- **Acteur** : représente le rôle joué par quelque chose ou quelqu'un se trouvant dans l'environnement du système étudié.
- **Auxiliaire** : classe qui supporte une autre classe centrale ou fondamentale, généralement en implémentant un flux logique ou de contrôle secondaire.
- **Classe d'implémentation** : permet de caractériser les classes qui sont nécessaires à l'implémentation physique du système.
- **Classe Meta** : classe dont les instances sont des classes. En règle générale, les méta-classes sont utilisées pour construire des méta-modèles.
- **Contrôle** : permet de caractériser les classes effectuant des traitements internes au système. Ceux-ci nécessitent généralement le concours de plusieurs classes.
- **Entité** : permet de caractériser des classes passives qui ne génèrent aucune interaction par elles-même. Elles peuvent participer à plusieurs cas d'utilisation et survivent généralement à une interaction unique. Elles représentent des objets partagés entre les différents acteurs qui les manipulent.
- **Enumération** : type de données contenant une liste de valeurs tabulées.
- **Expression** : expressions de types de données complexes basés sur des types.
- **Focus** : classe qui définit le flux logique ou de contrôle principal pour la ou les classes auxiliaires qui la supportent.
- **Frontière** : permet de caractériser les classes qui sont en prise directe avec l'environnement du système. Les interfaces hommes-machines en font partie.
- **Interface** : une interface est constituée d'un ensemble d'opérations qui décrivent le comportement d'un élément. En particulier, une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur.

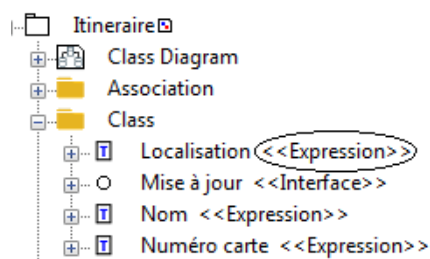
☛ *Ce sont des interfaces entre les différents composants du système informatique. Ce ne sont pas des interfaces avec les utilisateurs du système qui sont du stéréotype frontière. Voir "[Spécifier les interfaces](#)", [page 192](#) pour plus de détails.*

- **Opérateur** : représente un acteur humain qui interagit avec le système. Un opérateur interagit avec d'autres opérateurs et manipule des entités lorsqu'il participe à la réalisation d'un cas d'utilisation.
- **Opérateur externe** : un opérateur externe interagit directement avec des acteurs extérieurs au système.
- **Opérateur interne** : un opérateur interne interagit avec d'autres opérateurs et entités à l'intérieur du système.
- **PowerType** : méta-type dont les instances sont des sous-types d'un autre type.
- **Structure** : classe servant à décrire une structure utilisée dans les programmes.
- **Thread** : stéréotype utilisé dans l'implémentation d'un objet actif en tant que processus léger.
- **Type élémentaire** : permet de caractériser les types de données.

- **Utilitaire** : une classe de ce stéréotype regroupe des variables globales et des procédures utiles à la programmation décrites sous forme des attributs et opérations de cette classe.
- **Schema group** : classe décrivant un type d'élément XML, dont les sous-éléments forment un groupe.
- **XML Document Definition Root** : classe servant à décrire la structure d'un message échangé entre deux systèmes en utilisant la syntaxe du langage XML.

Option d'affichage des stéréotypes

Une option vous permet d'afficher les stéréotypes dans la fenêtre de navigation des objets.



Pour activer cette option :

1. Dans l'espace de travail **HOPEX**, cliquez sur **Menu principal > Paramètres > Options**.
2. Dans la partie gauche de la fenêtre des options, sélectionnez le dossier **Espace de travail**.
3. Dans la partie droite, cochez l'option **Afficher le stéréotype des objets UML dans le navigateur**.
4. Cliquez sur **OK**.

LES ATTRIBUTS

- ✓ "Définition d'un attribut", page 160
- ✓ "Spécifier les attributs d'une classe", page 161
- ✓ "Propriétés des attributs", page 161

Définition d'un attribut

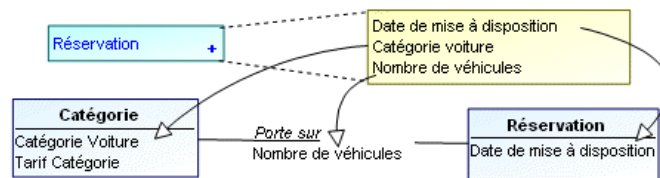
Un attribut est une propriété nommée d'une classe. C'est la donnée élémentaire mémorisée dans le système d'information de l'entreprise.

Exemples :

"Nom du client" (Attribut de la classe client).

"No client" (Identifiant de la classe client).

"Solde du compte" (Attribut de la classe compte).

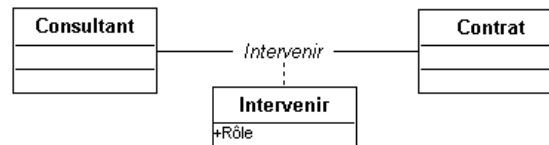


Les classes et les classes d'associations peuvent être caractérisées par des attributs.

Ces attributs ont pu, par exemple, être révélés par l'étude du contenu des messages qui circulent à l'intérieur de l'entreprise.

Un attribut est porté par une association quand sa valeur dépend de l'ensemble des classes participant à cette association.

Dans le diagramme présenté ci-après, le "rôle" qu'un "Consultant" a joué sur un "Contrat" dépend du consultant et du contrat, donc de l'association "Intervenir".



Spécifier les attributs d'une classe

Pour définir les attributs d'une classe :

1. Sélectionnez la classe et affichez ses propriétés.
2. Dans la fenêtre des propriétés, cliquez sur la liste déroulante puis sur **Composants**.
3. Sous la section **Attributs**, cliquez sur **Nouveau**.
Le nouvel attribut apparaît.
4. Cliquez sur le nom pour le modifier.

Pour chaque attribut, il est possible de préciser :

- Son **Type**, qu'il est possible d'exprimer sous forme d'une expression.

Ex : Integer.

☛ L'expression doit être conforme à la syntaxe UML. Voir "[Signature d'une opération ou d'un signal](#)", page 164 pour plus de détails.

☛ Voir aussi : "[Type d'un attribut](#)", page 162.

- Sa **Visibilité** :
 - "Public": c'est la visibilité par défaut. L'attribut est visible par tous.
 - "Protégé": l'attribut est visible par les héritiers de son paquetage ou ses amis.
 - "Privé": l'attribut est visible par sa classe ou ses *amis*.

📖 Les amis d'une classe sont des classes autorisées à connaître les internes de cette classe. Il est possible de préciser les amis d'une classe dans l'onglet Compléments de la fenêtre de propriétés de la classe.

- Sa **Multiplicité**, c'est à dire le nombre de répétitions de cet attribut dans la classe.

Attributs hérités


Lorsqu'une généralisation existe entre une classe générale et une classe particulière, la classe particulière hérite des attributs de la classe générale.

- ☛ Cliquez sur le bouton **Attributs hérités** pour visualiser les attributs hérités d'autres classes.

Propriétés des attributs

Pour ouvrir la fenêtre de propriétés d'un attribut :

1. Dans la fenêtre de propriétés de la classe détentrice, sélectionnez la page **Composants**.
2. Sous la section **Attributs** sélectionnez l'attribut en question.
3. Cliquez sur **Propriétés**.

☛ Le bouton  permet d'afficher les commandes cachées.

Dans la page **Caractéristiques**, vous pouvez préciser :

- Le **Type** de l'attribut sous la forme d'une **expression** (voir "[Type d'un attribut](#)", page 162).
- S'il s'agit d'un attribut **statique** : indique si l'attribut peut prendre des valeurs spécifiques pour chacune des instances de la classe ou bien avoir une valeur qui caractérise l'ensemble de la classe.
 - "Oui" : l'attribut a une valeur qui caractérise l'ensemble de la classe. Par exemple, l'attribut "Longueur des numéros de téléphone" de la classe "Client France" est de 10 chiffres.
 - "Non" : l'attribut peut prendre une valeur différente pour chacune des instances de la classe. Par exemple, l'attribut "Numéro de téléphone" prend une valeur différente pour chaque instance de la classe "Client".
- S'il s'agit d'un attribut **Persistant**, c'est-à-dire si la valeur de cet attribut doit être conservée dans le temps ou si elle ne vit que le temps du traitement en mémoire dans l'ordinateur.
- Sa **Multiplicité**, c'est-à-dire le nombre de répétitions de cet attribut dans la classe.
- S'il est en **Lecture seule**, c'est-à-dire si sa valeur peut être modifiée après avoir été renseignée une première fois.
- S'il s'agit d'un **Attribut Calculé**, c'est-à-dire que sa valeur est déduite de la valeur d'un ou plusieurs autres attributs.
- La **Valeur initiale** de l'attribut, qu'il prendra lors de la création d'une instance de la classe.

Type d'un attribut

Un type de données définit la nature des valeurs que peut prendre une donnée. Il peut être simple (entier, caractère, text, booléen, date...) ou plus élaboré et composé.

Un type est implémenté sous forme de classe.

Toute classe peut être utilisée pour typer un attribut ou un paramètre.

Exemple : Client, Commande, Fenêtre, Tableau

Les classes de stéréotype "Type élémentaire" sont créées uniquement afin de typer des attributs ou des paramètres. Elles sont invariables.

Exemples de types élémentaires :

Chaîne.

Entier.

Adresse Export.

Montant en devises.

Vous pouvez lister les types existants ou en créer de nouveaux.

☛ Les types proposés sont les classes détenues ou utilisées par le paquetage courant.

☛ Pour renseigner la structure d'un type, posez la classe correspondante dans le même diagramme ou dans un autre diagramme et activez la commande Propriétés de son menu contextuel.

LES OPÉRATIONS

- ✓ "Définition d'une opération", page 163
- ✓ "Spécifier les opérations d'une classe", page 163
- ✓ "Propriétés d'une opération", page 164
- ✓ "Signature d'une opération ou d'un signal", page 164
- ✓ "Paramètres d'une opération", page 166
- ✓ "Méthodes d'une opération (comportement opaque)", page 166
- ✓ "Le diagramme d'objets", page 197
- ✓ "Exceptions d'une opération", page 167
- ✓ "Afficher les attributs et les opérations d'une classe", page 167

Définition d'une opération

Une opération est un service qui peut être demandé à un objet pour mettre en oeuvre un comportement défini. Une opération possède une signature qui permet de préciser les paramètres qui lui sont nécessaires.

Exemples :

"Calcul Age" (Opération de la classe client).

"Imprimer" (Opération de la classe dessin).

"Calculer échéancier" (Opération de la classe prêt).

➤ Les opérations ne sont pas prises en compte par les outils de **HOPEX Information Architecture** (synchronisation, génération etc.).

Spécifier les opérations d'une classe

Pour spécifier les opérations d'une classe :

1. Sélectionnez la classe et affichez ses propriétés.
2. Dans la fenêtre des propriétés, cliquez sur la liste déroulante puis sur **Composants**.
3. Sous la section **Opérations**, cliquez sur **Nouveau** pour créer une opération ou sur **Relier** pour relier une opération existante.

L'opération apparaît dans les propriétés de la classe.

Vous pouvez préciser sa signature.

Opérations héritées


Lorsqu'une généralisation existe entre une classe générale et une classe particulière, la classe particulière hérite des opérations de la classe générale.

- Cliquez sur le bouton **Opérations héritées** pour visualiser les opérations héritées d'autres classes.

Propriétés d'une opération

Pour ouvrir la fenêtre de propriétés d'une opération :

1. Dans la fenêtre de propriétés de la classe détentrice, sélectionnez la page **Composants**.
2. Sous la section **Opérations**, sélectionnez l'opération en question.
3. Cliquez sur **Propriétés**.


 Le bouton permet d'afficher les commandes cachées.

Vous pouvez indiquer pour chaque opération :

- Son **Stéréotype** afin de préciser son utilisation :
 - **Constructeur** : créer une instance de la classe.
 - **Destructeur** : détruire une instance de la classe.
 - **Itérateur** : parcourir les instances de la classe.
 - **Sélecteur** : effectuer une sélection parmi les instances de la classe.
- S'il s'agit d'une opération **Statique** : si l'opération qui peut prendre des valeurs spécifiques pour chacune des Instances de la classe ou bien avoir une valeur qui caractérise l'ensemble de la Classe.
- La **Concurrence** permet de préciser le comportement de l'opération lorsqu'elle est appelée plusieurs fois simultanément.
 - **Concurrente** : l'opération répond simultanément aux différents appels.
 - **Protégée** : l'opération répond au premier appel et rejette les suivants.
 - **Séquentielle** : l'opération répond successivement à chacun des appels.
- Si c'est une opération en **Consultation**, c'est-à-dire qui ne modifie pas l'état de l'objet.
- Si cette opération est **Polymorphe**, c'est-à-dire que des méthodes peuvent être redéfinies pour cette opération dans des sous-classes.

Les indications suivantes sont utilisées pour compléter la signature de l'opération.

- Le **Type expression** de l'opération.

 Le type expression d'une opération précise le type de la variable retournée par l'opération à la fin de son exécution.

- Sa **Signature**.

Signature d'une opération ou d'un signal

La signature d'une opération ou d'un signal est constituée du nom de l'opération (du signal), de son type de retour, et de ses paramètres avec leurs types. La syntaxe UML standard est utilisée pour cette signature, elle est du type : Ope0 (Param0 : M-Bool) : M-Bool.

Il est possible de définir la signature :

- Soit dans la fenêtre de propriétés de l'opération ou du signal.
- Soit dans la fenêtre de propriétés de la classe à laquelle appartient l'opération, sous la section des **Opérations**.



La signature qui est conservée inclut une référence au type : si le type est renommé, les signatures qui l'utilisent reflètent cette évolution.

Syntaxe des signatures

La syntaxe standard des signatures est :

```
nomopération(paramètre1:expressiondutype1,paramètre2:expres  
siondutype2,...):expressiondutyperetour
```

Les noms comportant des blancs ou des caractères spéciaux doivent être placés entre apostrophes ('Nom du client'). Lorsqu'un nom contient une apostrophe, il faut dupliquer l'apostrophe : 'Nom de l' 'acheteur'

Exemples de signatures :

```
Déstocker (Produit0 : Entier(3), Quantité0 : Entier) :  
Booléen  
'Création de commande' ('Nom du client' : Client) : Byref  
Variable
```

Dans la spécification d'une signature, il est possible de préciser le paquetage auquel appartient une classe, séparé par deux fois deux points.

Exemple : `Entreprise::'Gestion Commerciale'::Client.`

La classe citée est reliée au paramètre ou au type de retour. Si elle n'existe pas, elle est créée. De même, les paquetages cités dans le parcours qui n'existent pas sont créés et reliés à la classe.

Si le paquetage n'est pas précisé, une fenêtre vous propose de choisir entre les éventuelles classes homonymes.

Paramètres d'une opération

Un paramètre est la spécification d'une variable qui peut être modifiée, transmise ou retournée. Un paramètre peut préciser un nom, un type et une direction. Les paramètres sont utilisés pour les opérations, les messages et les événements.

Un argument est une valeur spécifique correspondant à un paramètre.

Dans la fenêtre de propriétés d'une opération, la section **Paramètres** vous permet de préciser :

- Son **Type** sous forme d'une expression Ex : Integer(5).
- Sa **Valeur par défaut** Ex : 0.
- Sa **Direction** : en entrée et/ou en sortie de l'opération.

Pour créer un *paramètre* sur une opération :

1. Affichez les propriétés de l'opération.
2. Sélectionnez la page **Caractéristiques**.
3. Sous la section **Paramètres**, cliquez sur **Nouveau**.
La fenêtre de création d'un paramètre apparaît.
4. Saisissez le nom du paramètre et cliquez sur **OK**.

Méthodes d'une opération (comportement opaque)

Une méthode - ou comportement opaque - est une représentation textuelle de l'implémentation d'une opération, d'une classe ou d'un composant. Elle spécifie l'algorithme ou la procédure qui produit les résultats d'une opération ou le comportement d'un élément.

Pour définir la méthode qui implémente une opération :

1. Ouvrez la fenêtre de propriétés de l'opération.
2. Sélectionnez la page **Caractéristiques**.
3. Sous la section **Méthode**, cliquez sur **Ajouter**.
La fenêtre d'ajout d'une méthode apparaît.
4. Saisissez le nom de la méthode à créer ou recherchez une méthode existante.
5. Cliquez sur **OK**.

Pour saisir le corps du texte de la méthode qui implémente l'opération :

1. Ouvrez la fenêtre de propriétés de la méthode.
2. Sélectionnez la page **Caractéristiques**.
3. Définissez la méthode dans le cadre **Corps**.

Quand une classe possède plusieurs sous-classes, chaque sous-classe est susceptible d'implémenter l'opération sous forme d'une méthode différente.

La section **Méthode** présente la méthode relative à la classe sélectionnée.

Conditions d'une opération

Vous pouvez définir les conditions d'une opération sous la forme de contraintes.

Les types de condition sont :


- La **Pré-Condition**, qui doit être vérifiée avant que l'opération ne s'exécute.
- La condition sur le **Corps**, qui doit être vérifiée lors de l'exécution de l'opération.
- La **Post-Condition** qui doit être vérifiée après l'exécution de l'opération.

Pour saisir une condition sur une opération :

1. Ouvrez la fenêtre de propriétés de l'opération.
2. Sélectionnez la page **Conditions**.
3. Sous la section **Conditions**, sélectionnez le type de condition :
 - pré-condition
 - condition sur le corps
 - post-condition
4. Cliquez sur **Nouveau**.
La fenêtre d'ajout d'une contrainte apparaît.
5. Saisissez le nom de la contrainte à créer ou recherchez une contrainte existante.
6. Cliquez sur **OK**.

Pour saisir le corps de la condition :

1. Dans la fenêtre de propriétés de l'opération détentrice, sélectionnez la condition.
2. Cliquez sur le bouton **Propriétés**.

☛ Le bouton  permet d'afficher les commandes cachées.

3. Cliquez sur la page **Caractéristiques**.
4. Sous la section **Corps expression**, saisissez l'expression de la condition.

Exceptions d'une opération

Si une condition n'est pas respectée, une exception est générée.

La section **Exceptions** vous permet de définir les messages d'erreur envoyés par l'opération lorsqu'une exception se produit et de préciser leur signature.

Afficher les attributs et les opérations d'une classe

Pour modifier l'affichage des attributs et des opérations de la classe :

1. Faites un clic droit sur la ou les classes dont vous voulez afficher des attributs.
2. Sélectionnez **Formes et détails**.
La fenêtre de sélection des éléments à afficher apparaît.
3. Dans l'arbre de gauche, cliquez sur **Attribut**.

4. Sélectionnez les attributs que vous voulez afficher.

Vous pouvez les afficher **Tous**, en afficher **Certains** que vous allez sélectionner dans la liste, ou n'en afficher **Aucun**.

Vous pouvez demander l'affichage de la **Visibilité**, du *type*, ... de chacun des attributs.



Un type de données permet de mettre en commun des caractéristiques communes à plusieurs attributs. Les types de données sont implémentés sous forme de classe.



Vous pouvez cacher ou rendre visible le compartiment présentant les attributs dans le dessin de la classe, en activant ou désactivant la case à cocher devant "Affichage de".

L'affichage des opérations est indiqué de la même manière, en choisissant **Opération** dans l'arbre qui vous est présenté.

LES SIGNAUX

Définition d'un signal

Un signal est un événement qui peut être invoqué explicitement. Un signal peut posséder des paramètres. Un signal peut être envoyé à un objet ou à un ensemble d'objets. Il peut être invoqué dans le cadre de la participation d'un acteur à un cas d'utilisation.

Un signal peut être émis ou reçu par une classe. Il peut être également émis par une opération à la suite d'une exception.

Spécifier les signaux d'une classe

Créer un signal émis ou reçu



Pour préciser quels sont les signaux qui peuvent être émis ou reçus par une classe :

1. Sélectionnez la classe et affichez ses propriétés.
2. Dans la fenêtre de propriétés, cliquez sur la liste déroulante puis sur **Compléments**.
3. Dans l'arbre qui vous est présenté, sélectionnez la branche **Signal émis** ou **Signal reçu**, puis cliquez sur le bouton **Ajouter**.
4. Indiquez le nom du signal et cliquez sur **OK**.

Propriétés d'un signal

Pour ouvrir la fenêtre de propriétés d'un signal :

1. Dans la fenêtre de propriétés d'une classe, dans la page **Compléments**, sélectionnez le signal et cliquez sur **Propriétés**.

 Le bouton  permet d'afficher les commandes cachées.

La fenêtre de propriétés du signal apparaît.

Vous pouvez indiquer pour un signal :

- Son **Stéréotype** afin de préciser son utilisation :
 - **Exception** : signal d'erreur généré lorsqu'une exception se produit pendant l'exécution d'une opération.
- Sa **Visibilité** par rapport au paquetage :
 - **Publique** : c'est la visibilité par défaut. Le signal est visible par tout élément situé à l'extérieur du paquetage.
 - **Protégée** : le signal est visible par les éléments héritiers ou amis.
 - **Privée** : le signal est visible par sa classe ou ses amis.



Les amis d'une classe sont des classes autorisées à connaître les internes de cette classe. Il est possible de préciser les amis d'une classe dans l'onglet Compléments de la fenêtre de propriétés de la classe.

- Le **Type expression** du signal (voir *type expression*).



Le type expression d'un signal précise le type de la variable retournée par le signal lors de sa réception par son destinataire.

Un signal peut être une demande de **Vote** envoyé à chacun des objets actifs pour leur demander s'il est possible d'effectuer une action, comme par exemple fermer une session Windows.

Un signal peut donner lieu à une **Diffusion** générale à tous les objets actifs.

Paramètres d'un signal

Les *Paramètres* du signal sont renseignés sous l'onglet **Paramètres** de sa fenêtre de propriétés. Vous pouvez préciser :

- Son **Type** sous forme d'une expression. Ex : Integer(5).
- Sa **Valeur** par défaut. Ex : 0.
- Sa **Direction** : en entrée et/ou en sortie de l'opération.



Un paramètre est la spécification d'une variable qui peut être modifiée, transmise ou retournée. Un paramètre peut préciser un nom, un type et une direction. Les paramètres sont utilisés pour les opérations, les messages et les événements.



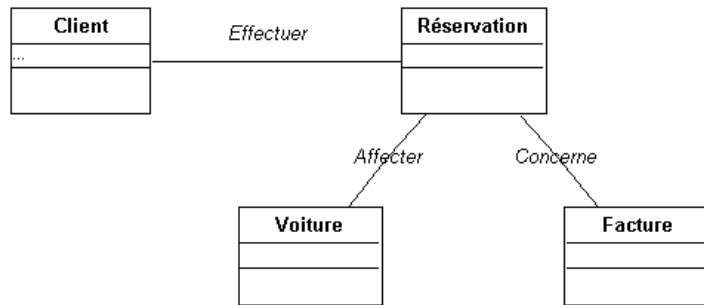
Un argument est une valeur spécifique correspondant à un paramètre.

LES ASSOCIATIONS

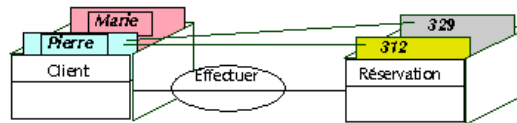
Une association est une relation existant entre deux classes.

Une association est binaire quand elle relie deux classes, ternaire quand elle en relie trois, etc.

Les associations peuvent être comparées à des liens entre des fiches.



Le dessin suivant permet de visualiser "en trois dimensions" les situations qu'un diagramme de classes permet de mémoriser.



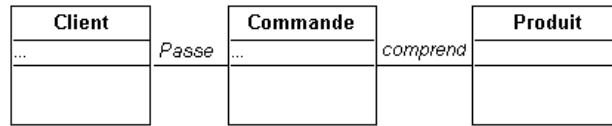
Pierre et Marie sont des clients. Pierre a effectué les réservations numéros 312 et 329.

Un diagramme de classes doit permettre de mémoriser toutes les situations du contexte de l'entreprise.

☛ Le diagramme ne doit pas permettre de représenter des situations irréalistes ou aberrantes.

Exemples d'association

- Un client *pass*e une commande.
- Une commande *comprend* plusieurs produits.



- Une personne *travaille* pour une société.






- Une alarme *est déclenchée* par un capteur.

Un capteur *couvre* une zone.

Une fenêtre *affiche* une chaîne de caractères.

Créer une association

Pour créer une *association*:

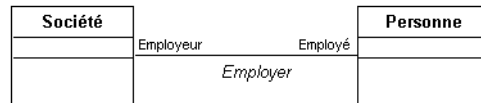
1. Dans le diagramme de classes, cliquez sur le bouton **Association**  de la barre d'objets.
2. Cliquez sur une des classes concernées, et faites glisser la souris jusqu'à la deuxième classe avant de relâcher votre pression.
La fenêtre de création d'une association apparaît.
3. Saisissez le nom de l'association à créer.
 Vous pouvez également sélectionner une association existante.
4. Cliquez sur **Ajouter**.
L'association apparaît dans le diagramme sous forme d'un trait.
 En cas d'erreur, vous pouvez supprimer un élément ou un lien en cliquant avec le bouton droit sur cet élément, et en sélectionnant la commande *Supprimer* dans son menu contextuel.

Les rôles des associations

Il est possible de décrire les différents *rôles* joués par les classes dans les associations, et de préciser leur multiplicité et leur navigabilité.

Chaque extrémité d'une association permet de préciser le rôle joué par chaque classe dans l'association.

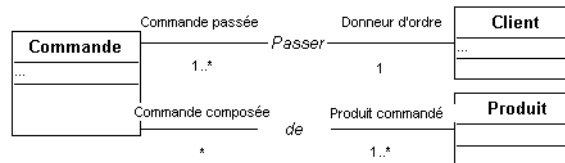
Visuellement, le nom du rôle se distingue du nom d'une association, car il est placé près de son extrémité. De plus, il apparaît en caractères droits, alors que le nom de l'association est en italique.



Lorsque deux classes sont reliées par une seule association, le nom des classes suffit souvent à caractériser le rôle ; nommer les rôles prend tout son intérêt lorsque plusieurs associations relient deux classes.

Exemples de rôles :

- Un client est le *donneur d'ordre* d'une commande.
- Une commande est *passée* par un client.
- Une commande est *composée* de produits.
- Un produit est *commandé*.



Une personne est un *employé* dans une société.

Une société est *l'employeur* de ces personnes.

Une alarme est *déclenchée* par un ou plusieurs capteurs.

Une zone est *couverte* par un capteur.

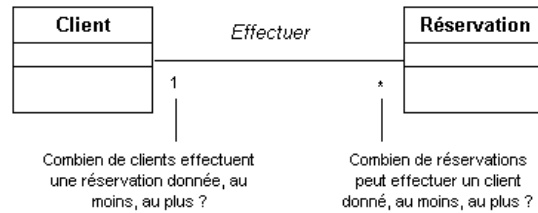
Une ou plusieurs chaînes de caractères sont *affichées* dans une fenêtre.

Multiplicité d'un rôle

La multiplicité précise l'intervalle entre les valeurs minimum et maximum des cardinalités possibles pour un ensemble. On l'indique en particulier pour chacun des rôles que jouent les classes dans une association. Elle peut prendre les valeurs *, 0..1, 1, 1..*, 2..*, 4..10, etc. La valeur proposée par défaut est *.

La cardinalité est le nombre d'éléments contenus dans un ensemble.

La **multiplicité** exprime le nombre minimum et maximum d'instances d'une classe pouvant être reliées par l'association à chaque instance de l'autre classe.

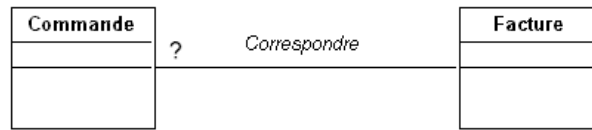


Les multiplicités usuelles sont "1", "0..1", "*" ou "0..*", "1..*", et "M..N" où "M" et "N" sont des entiers :

- La multiplicité "1" indique qu'une et une seule instance de la classe est reliée par cette association à chaque instance de l'autre classe.
- La multiplicité "0..1" indique qu'au plus une instance de la classe peut être reliée par cette association à chaque instance de l'autre classe.
- La multiplicité "*" ou "0..*" indique qu'un nombre quelconque d'instances de la classe peuvent être reliées par l'association à chaque instance de l'autre classe.
- La multiplicité "1..*" indique qu'au moins une instance de la classe est reliée par l'association à chaque instance de l'autre classe.
- La multiplicité "M..N" indique qu'au moins M instances de la classe et au plus N sont reliées par l'association à chaque instance de l'autre classe.

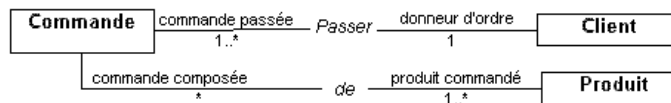
1	Un et un seul
0..1	Zéro ou un
M..N	De M à N (entiers naturels)
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	De un à plusieurs

L'exemple suivant va nous permettre d'illustrer la signification de chacune des multiplicités.



- 0..1 : A une commande correspond une facture au maximum ou aucune.
- * : Aucune restriction n'est imposée sur le nombre de factures correspondant à une commande.
- 1 : A chaque commande correspond une facture et une seule.
- 1..* : A chaque commande correspond une ou plusieurs factures.

Autres exemples de multiplicité :



- 1..* : Un client peut passer une ou plusieurs commandes.
- 1 : Une commande est passée par un et un seul client.
- 1..* : Une commande comprend un ou plusieurs produits.
- * : Un produit peut faire partie de plusieurs commandes, ou d'aucune.
- 0..1 : Une personne travaille pour une société.
- 1..* : Une alarme est déclenchée par un ou plusieurs capteurs.
- 1 : Un capteur couvre une et une seule zone.
- 1..* : Une fenêtre affiche une ou plusieurs chaînes de caractères.

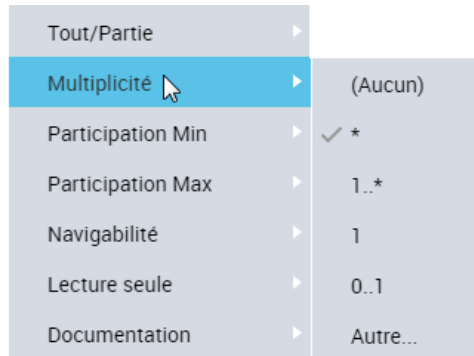
Préciser la multiplicité d'un rôle

Pour préciser la multiplicité d'un rôle :

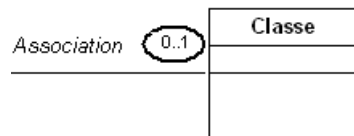
1. Faites un clic droit sur la partie du trait de l'association qui se trouve au plus près de la classe.

2. Sélectionnez **Multiplicité** puis la valeur voulue.

☛ Si le menu affiché ne propose pas les multiplicités, vérifiez que vous avez bien cliqué sur le trait qui matérialise le rôle, et non sur l'association.



La multiplicité apparaît alors sur le rôle.



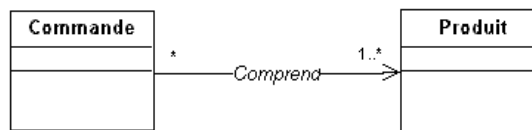
😊 Toutes les informations précisées avec le menu contextuel peuvent être consultées et modifiées dans la fenêtre de propriétés du rôle.

Navigabilité d'un rôle

La navigabilité précise le sens dans lequel l'association entre deux classes peut être parcourue. Pour ne pas encombrer le dessin, on n'indiquera la navigabilité que lorsqu'elle n'a lieu que dans un seul sens.

Exemple de navigabilité :

- Il est nécessaire de connaître tous les produits contenus dans une commande.
- Par contre, il est rarement utile de connaître toutes les commandes qui portent sur un produit.



Préciser la navigabilité d'un rôle

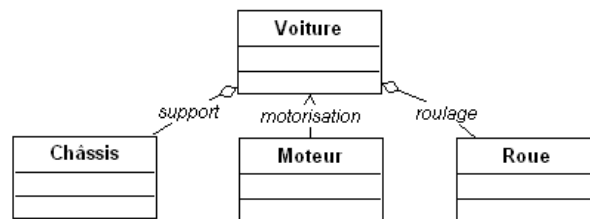
Pour indiquer qu'une association n'est navigable que dans un sens :

1. Faites un clic droit sur le rôle non navigable.
2. Sélectionnez **Navigabilité > Non**.
Une flèche représentant la navigabilité apparaît alors sur le rôle opposé.

Agrégation d'un rôle

L'agrégation est une forme particulière d'association qui indique que l'une des classes contient l'autre.

Exemple : Une voiture comprend un châssis, un moteur et des roues.



Préciser l'agrégation d'un rôle

Pour préciser l'agrégation d'un rôle :

1. Faites un clic droit sur le rôle.
2. Sélectionnez **Tout/Partie > Agrégat**.

☛ Si le menu affiché ne propose pas l'agrégation, vérifiez que vous avez bien cliqué sur le trait qui matérialise le rôle, et non sur l'association.

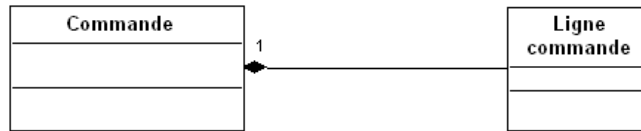
Un losange représentant l'agrégation apparaît alors sur le rôle.

Composition d'un rôle

La composition est une agrégation forte pour laquelle la durée de vie des composants coïncide avec celle du composé. Une composition est une agrégation immuable avec une multiplicité 1.

Exemple : Une commande est composée de plusieurs lignes de commande qui n'existent plus si la commande est supprimée.

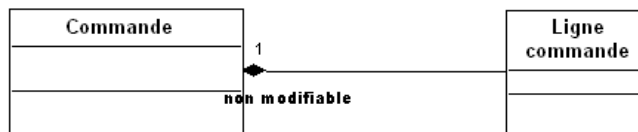
La composition est matérialisée par un losange noir.



Rôle modifiable

La caractéristique **Lecture seule** permet de préciser si le rôle joué par une classe dans une association est modifiable après qu'il a été créé ou non. Par défaut, le rôle d'une classe dans une association est considéré comme modifiable.

Exemple : Une commande comprend une ligne de commande pour chacun des produits commandés. On ne peut plus changer ces différentes lignes de commande après l'enregistrement de la commande.



Il est possible d'indiquer si un rôle est modifiable à l'aide du menu contextuel du rôle ou dans la fenêtre de propriétés du rôle.

La caractéristique **Lecture seule** du rôle peut avoir les valeurs suivantes :

- **Ajout seul** : il est toujours possible de relier de nouveaux objets par cette association, mais il n'est pas possible de délier les objets déjà reliés.
- **Lecture seule** : les instances reliées ne peuvent plus être déliées. Il n'est pas possible non plus d'ajouter un nouveau lien.
- **Pas de restriction** : de nouvelles instances peuvent être reliées ou déliées à tout moment sans aucune contrainte.

Ordre d'un rôle

Il est possible de préciser si un rôle est ordonné ou non. Par exemple, pour une commande, il peut être intéressant de mémoriser l'ordre de ses lignes de commande.

Pour spécifier qu'un rôle est ordonné :

1. Ouvrez la fenêtre de **Propriétés** du rôle.
2. Dans la page **Caractéristiques**, cochez la case **Ordonnée**.

Propriété statique d'un rôle

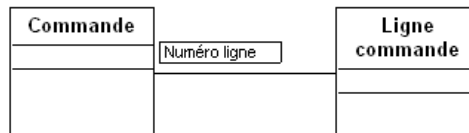
Comme pour un attribut, il est possible de préciser si un rôle peut prendre des valeurs spécifiques pour chacune des instances de la classe ou bien s'il a une valeur qui caractérise l'ensemble de la classe :

1. Ouvrez la fenêtre de propriétés du rôle.
2. Cliquez sur la page **Caractéristiques**.
3. Dans le champ **Statique**, sélectionnez :
 - "Oui" : pour que le rôle ait une valeur qui caractérise l'ensemble de la classe.
 - "Non" : pour que le rôle puisse prendre une valeur différente pour chacune des instances de la classe.

Qualificatif d'un rôle

Un qualificatif est un attribut dont les valeurs partitionnent l'ensemble des objets reliés à un objet à travers une association.

Exemple : Une commande comprend plusieurs lignes de commande. Le numéro de ligne de commande peut servir de qualificatif pour identifier chacune des lignes.

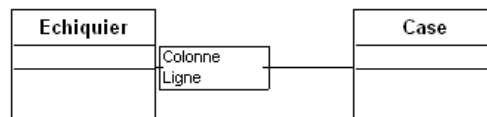


Pour définir un qualificatif :

1. Faites un clic droit sur le rôle et sélectionnez **Propriétés**.
La fenêtre de propriétés du rôle s'ouvre.
2. Sélectionnez la page **Qualificatifs**.
3. Pour ajouter un nouveau qualificatif au rôle, cliquez sur **Ajouter**.
4. Saisissez le nom du qualificatif.
5. Cliquez sur **Ajouter**.

Plusieurs qualificatifs peuvent être nécessaires pour identifier de manière unique chacun des objets de la classe.

Par exemple, chaque case d'un échiquier est identifiée par son numéro de ligne et son numéro de colonne dans l'échiquier.



Surcharger un rôle

Un rôle peut hériter d'un rôle défini au niveau supérieur. La surcharge permet de définir des propriétés supplémentaires sur un rôle hérité.

Pour surcharger un rôle :

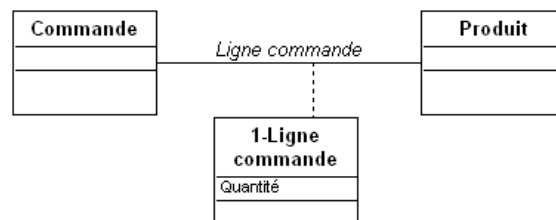
1. Ouvrez la fenêtre de propriétés du rôle.
2. Dans la fenêtre de propriétés, cliquez sur la liste déroulante puis sur **Caractéristiques**.
3. Sous la section **Rôles**, sélectionnez **Rôle surchargé**.
4. Cliquez sur **Ajouter**.
La fenêtre de sélection apparaît.
5. Recherchez et sélectionnez le rôle en question.
6. Cliquez sur **OK**.

Les classes d'association

Une classe d'association est une association qui possède aussi les propriétés d'une classe comme des attributs.

Il est utile de créer une classe d'association pour préciser des caractéristiques de l'association.

Par exemple, il est nécessaire de préciser la quantité de produit demandée pour chacune des lignes d'une commande.



Pour créer une classe d'association :

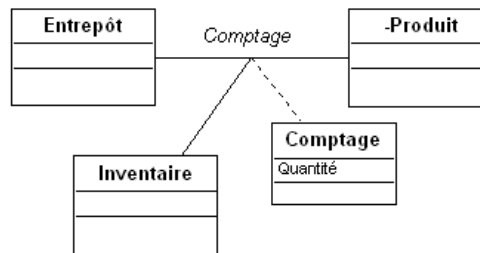
1. Créez une nouvelle classe.
2. A l'aide du bouton **Lien**, créez un lien entre la classe et l'association.
La classe d'association est reliée à l'association par un trait pointillé.

☛ Comme pour les classes standard, il est possible de cacher les compartiments et de retailler la classe d'association à l'aide de la commande *Affichage de son menu contextuel*.

Définir une association "plus que binaire"

Certaines associations associent non pas deux, mais davantage de classes. Ces associations sont, en principe, rares.

Exemple : Lors d'un inventaire, une certaine quantité de produit a été comptée dans chaque entrepôt.

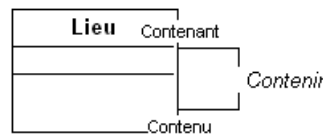


Pour créer une association ternaire :

1. Créez tout d'abord l'association entre deux des classes.
2. Cliquez sur le bouton **Rôle de l'association**
3. Tirez un lien entre l'association et la troisième classe.

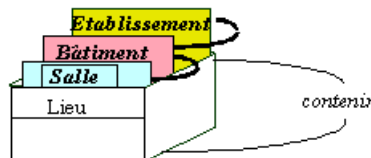
Vous pouvez ensuite créer la classe d'association éventuelle comme précédemment.

Les associations réflexives



Certaines associations mettent en jeu plusieurs fois la même classe.

Une salle de classe, un bâtiment, un établissement scolaire sont tous des lieux.




Une salle de classe est contenue dans un bâtiment, lui-même contenu dans un établissement scolaire.

Une association réflexive porte deux fois sur la même classe.

Créer une association réflexive

Pour créer une association réflexive :

1. Cliquez sur le bouton **Association**  de la barre d'outils.
2. Cliquez dans la classe concernée et faites glisser la souris en dehors de cette classe, puis revenez-y ; relâchez enfin votre pression.
L'association réflexive apparaît sous forme d'un crochet.

☛ Dans le cas d'une association entre une classe et elle-même, il est indispensable de préciser les rôles afin de distinguer les liens correspondants dans le dessin.

LES PARTIES

Dans un diagramme de classes, une partie représente un rôle joué par une instance d'une classe ou d'un composant lors de l'exécution d'une tâche.

Une partie appartient à une classe. L'appartenance est indiquée sur le lien de la partie.

Par exemple, ci-dessous, La Classe "Commande" est composée de la classe "Personne".

La partie est détenue par la classe "Commande" et référence la classe "Personne".



Créer une partie entre deux classes

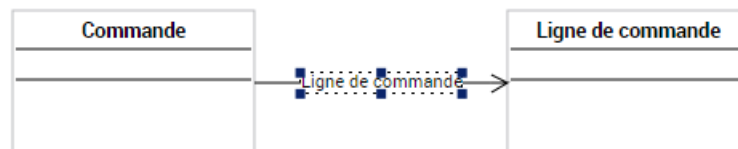
Une partie associe deux classes uniquement et elle est orientée.

Pour définir une partie entre deux classes :

1. Dans la barre d'objets du diagramme de classes, cliquez sur **Partie**.
2. Tirez un lien de la classe détentrice vers la classe référencée.
Le nom de la partie est défini automatiquement.

Définir l'identifiant d'une classe à travers une partie

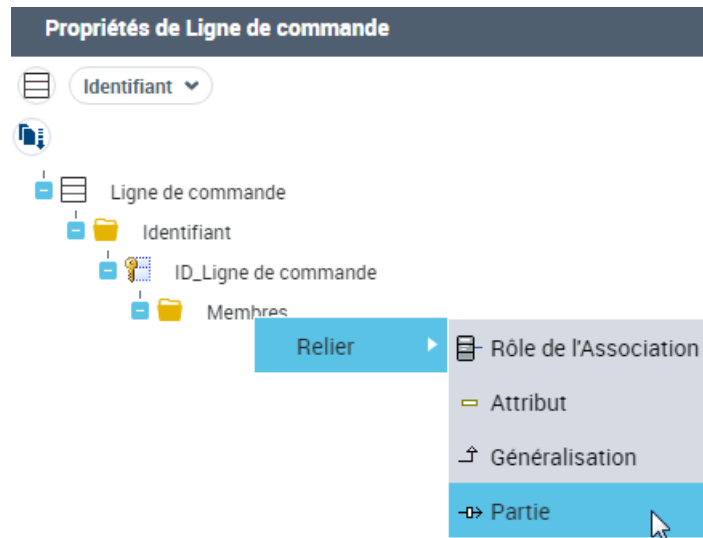
Dans l'exemple ci-dessous, l'identifiant de la classe "Ligne de commande" peut être défini à partir de la classe "Commande" à travers la partie "Ligne de commande".



Pour définir l'identifiant de la classe "Ligne de commande" :

1. Affichez les propriétés de la classe "Ligne de commande".

2. Sélectionnez la page **Identifiant**.
3. Faites un clic droit sur le dossier **Membres** et sélectionnez **Relier** > **Partie**.



4. Sélectionnez la partie proposée.
5. Cliquez sur **OK**.

Multiplicités des classes associées

A travers les multiplicités vous pouvez indiquer le nombre minimum et maximum d'instances des classes reliées par la partie.

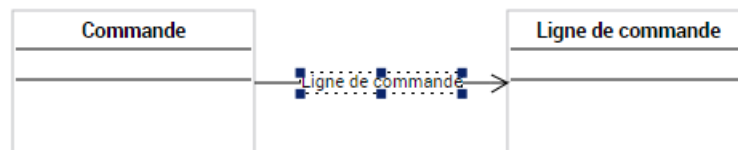
Exemple : 1 commande contient 1 ou plusieurs ligne(s) de commande.

Multiplicité de la classe référencée par la partie

La multiplicité de la classe référencée est à indiquer sur le lien de la partie.

Pour définir la multiplicité de la classe référencée :

1. Faites un clic droit sur le lien de la partie.

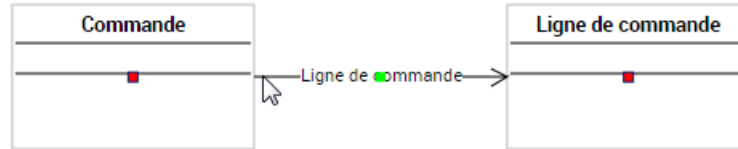


2. Sélectionnez **Multiplicité** puis la valeur souhaitée.

Multiplicité de la classe détentrice de la partie

Pour définir la multiplicité sur la classe détentrice de la partie :


1. Faites un clic droit sur le rôle de la partie associé à la classe détentrice.




2. Dans le menu contextuel qui apparaît, sélectionnez **Multiplicité** puis la valeur souhaitée.

Relations d'agrégation et de composition

Sur la partie qui relie deux classes vous pouvez définir une relation d'agrégation ou de composition.

 L'agrégation est une forme particulière d'association qui indique que l'une des entités contient l'autre.


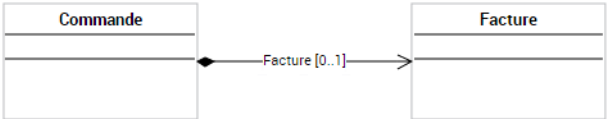

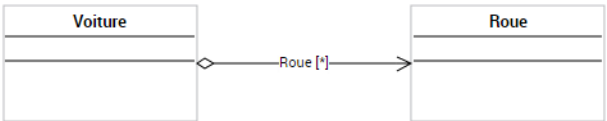

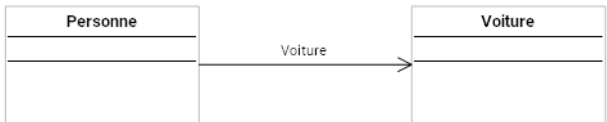
 La composition est une agrégation forte pour laquelle la durée de vie des composants coïncide avec celle du composé. Une composition est une agrégation immuable avec une multiplicité 1.

Pour définir un lien de composition ou d'agrégation entre les classes :

1. Faites un clic droit sur la partie.
2. Sélectionnez **Tout/Partie** puis la valeur souhaitée :
 - **Agrégat**
 - **Composé**

Multiplicités associées

Le tableau ci-dessous présente les multiplicités associées automatiquement aux agrégations et compositions.

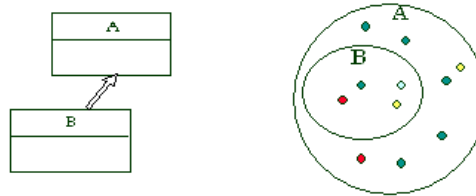
		Multiplicité correspondante	Exemple
Composition		1	 1, 1..*
Agrégation		0..1	 0..1, 1..*
Non renseigné		*	 *, *

LES GÉNÉRALISATIONS

Une généralisation représente une relation d'héritage entre une classe générale et une classe plus spécifique. La classe spécifique est cohérente avec la classe plus générale et en hérite ses caractéristiques et son comportement. Elle comporte cependant des informations supplémentaires. Toute instance de la classe spécifique est aussi une instance de la classe générale.

- ✓ "Qu'est-ce qu'une généralisation", page 187
- ✓ "Cas de plusieurs sous-classes", page 189
- ✓ "Intérêt des sous-classes", page 189
- ✓ "Héritage Multiple", page 190
- ✓ "Créer une généralisation", page 190
- ✓ "Discriminant", page 191

Qu'est-ce qu'une généralisation



La classe A est une généralisation de la classe B. Cela suppose que tous les objets de la classe B sont aussi des objets de la classe A. Autrement dit, B est un sous-ensemble de A.

B est alors la sous-classe, A la super-classe.

Exemple A : Personne, B : Parisien.

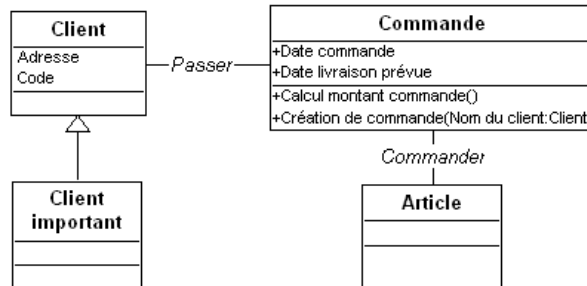
B étant un sous-ensemble de A, les objets de la classe B "héritent" des caractéristiques de ceux de la classe A.

Il n'est donc pas nécessaire de décrire de nouveau pour la classe B :

- Ses attributs
- Ses opérations
- Ses associations

Exemple

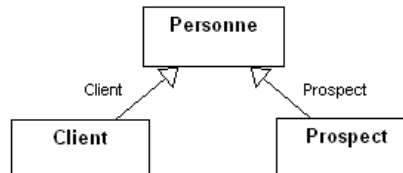
La classe "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 MF, peut être une spécialisation de la classe client (origine).



Dans l'exemple qui précède, les associations et les attributs spécifiés pour "Client" sont aussi valables pour "Client important".

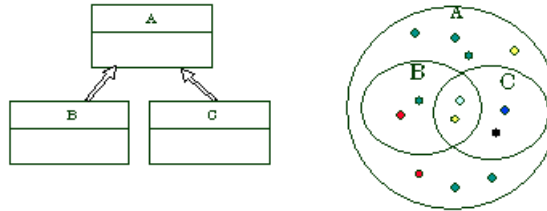
Autres exemples de généralisations :

- Prospect et client sont deux sous-classes de "personne".



- Commande export est une sous-classe de la classe "commande".
- Personne physique et personne morale sont deux sous-classes de la classe "personne".
- Polygone, ellipse et cercle sont des sous-classes de la classe "forme".
- Chêne, orme, et bouleau sont des sous-classes de la classe "arbre".
- Véhicule à moteur, véhicule tout-terrain et véhicule amphibie sont des sous-classes de la classe "véhicule".
- Camion est une sous-classe de la classe "véhicule à moteur".

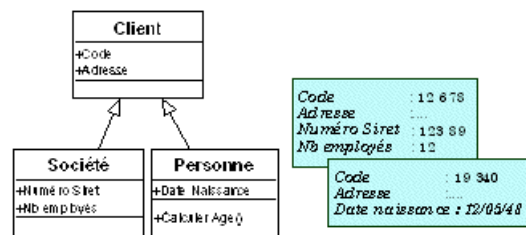
Cas de plusieurs sous-classes



Plusieurs sous-classes d'une même classe :

- Ne sont pas forcément exclusives.
- Ne forment pas nécessairement une partition.

Intérêt des sous-classes

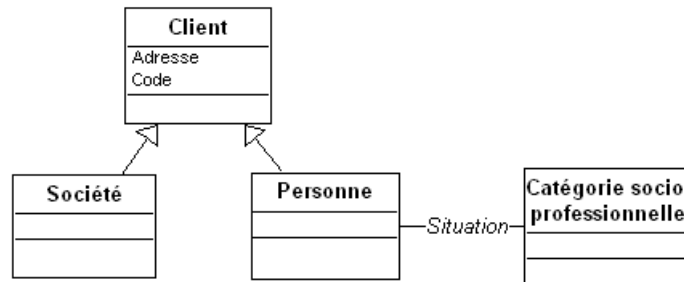


Une sous-classe hérite de tous les attributs, opérations et associations de sa super-classe, mais elle peut avoir des attributs ou des associations que ne possède pas sa super-classe.

Une sous-classe peut ainsi avoir des attributs spécifiques. Ceux-ci n'ont de sens que pour une sous-classe particulière. Dans l'exemple ci-dessus :

- Le "numéro de Siret" et le "nombre d'employés" n'ont de sens que pour une "société".
- La "date de naissance" est caractéristique d'une "personne", pas d'une "société".
- De même, il est utile de calculer l'"âge" d'une "personne". Cet attribut et cette opération n'ont généralement pas d'intérêt pour une "société".

Une sous-classe peut également avoir des *associations* spécifiques.



- Une "personne" entre dans une "catégorie socio-professionnelle" : "cadre", "employé", "commerçant", "agriculteur", etc. Cette classification n'a pas de sens pour une "société". (Il existe également une classification pour les entreprises, mais ce n'est pas la même que pour les personnes.)


Héritage Multiple

Il est parfois utile de spécifier une classe ayant plusieurs super-classes. La sous-classe hérite alors de toutes les caractéristiques des deux super-classes. Cette possibilité doit être utilisée avec précaution.

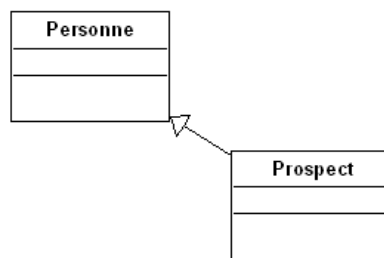
☛ L'héritage multiple n'est pas pris en compte pour la génération des tables.

Créer une généralisation

Pour créer une généralisation :

1. Cliquez sur le bouton **Généralisation**  de la barre d'outils
2. Cliquez dans la sous-classe et faites glisser la souris jusqu'à la super-classe, avant de relâcher votre pression.

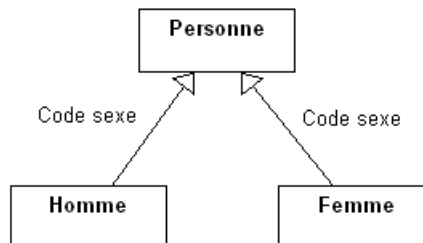
La généralisation apparaît dans le diagramme, matérialisée par une flèche.



Discriminant

Le discriminant est l'attribut d'une généralisation dont la valeur permet de répartir les objets entre les sous-classes associées à la généralisation.

Par exemple, l'attribut code-sexe permet de répartir les objets de la classe personne entre la sous-classe homme ou femme.



Vous pouvez définir le ou les discriminants dans la fenêtre de propriétés de la généralisation, sous la page **Discriminants**.

- ☛ Il est également possible de préciser si une généralisation est :
- *Disjointe* : Une instance ne peut pas appartenir simultanément à deux sous-classes de cette généralisation.
 - *Complète* : Toutes les instances de la super-classe appartiennent à au moins une des sous-classes de cette généralisation.

SPÉCIFIER LES INTERFACES


Une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur. L'interface est un stéréotype de classe.

Une interface est constituée d'un ensemble d'opérations qui décrivent le comportement d'un élément. En particulier, une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur.

Ce sont des interfaces entre les différents composants du système informatique.

Créer une interface

Pour créer une classe interface dans un diagramme de classes :

1. Dans la barre d'outils, sélectionnez le bouton **Interface** .
2. Cliquez sur le plan de travail du diagramme.
3. Dans la fenêtre qui apparaît, saisissez le nom de l'interface et cliquez sur le bouton **Ajouter**.
La classe interface apparaît dans le diagramme.

Vous pouvez spécifier les opérations de l'interface comme pour n'importe quelle classe.


Relier une interface à une classe


Lorsque vous reliez une classe à une interface, vous devez préciser s'il s'agit d'une interface requise ou d'une interface fournie par la classe.

Une interface requise est une interface nécessaire au fonctionnement de l'objet.

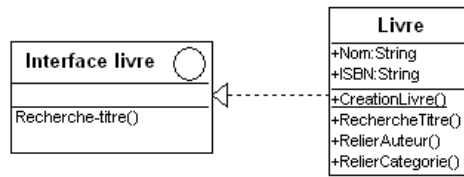
Une interface fournie est une interface mise à disposition par un objet à destination d'autres objets.

Pour relier une interface à une classe :

1. Cliquez sur le bouton **Lien** .
2. Effectuez le lien en partant de la classe pour aller vers l'interface. Une fenêtre apparaît.
3. Sélectionnez le type de lien à créer : interface fournie ou interface requise.

 D'autres types de liens, spécifiques aux classes, peuvent apparaître.



4. Cliquez sur **OK**.



SPÉCIFIER LES DÉPENDANCES

Dans le diagramme de classes, pour indiquer qu'un paquetage référence une classe ou un autre paquetage :

1. Cliquez sur le bouton **Lien** 
2. Effectuez le lien en partant d'un paquetage vers le paquetage ou la classe qu'il référence.

 Le bouton **Vues**  vous permet de préciser les boutons que vous voulez voir apparaître dans la barre d'objets.

SPÉCIFIER DES CLASSES PARAMÉTRÉES

Une classe paramétrée permet de définir des caractéristiques et un comportement modulable en fonction de la valeur de certains paramètres. Par exemple, une classe paramétrée peut être utilisée pour gérer des listes d'objets. Le paramètre sera alors le type d'objet que l'on veut gérer sous forme de liste. Ce type de classe est implémenté en particulier dans le langage C++.

Pour spécifier une classe paramétrée :

1. Ouvrez la fenêtre des **Propriétés** de la classe et sélectionnez la page **Caractéristiques**.
2. Vous pouvez saisir les paramètres et préciser éventuellement leur type.


Paramètres:

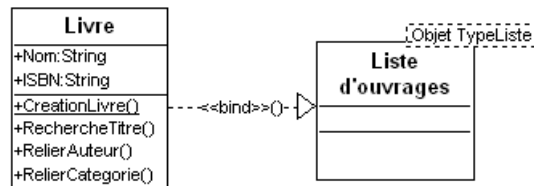


Objet TypeListe : int

Les paramètres de la classe s'affichent en haut à droite.

Pour relier une classe à une classe paramétrée :

1. Cliquez sur le bouton **Lien** .
2. Créez le lien en partant de la classe pour aller vers la classe paramétrée.



LES CONTRAINTES

Une contrainte est une déclaration qui établit un contrôle ou une règle de gestion impliquant généralement plusieurs classes.

La plupart des contraintes impliquent les associations entre les classes.


Exemples de contraintes :

- La personne responsable d'un service doit appartenir à ce service.
- Toute commande facturée doit avoir été livrée auparavant.
- La date de livraison doit être postérieure à la date de commande.

Un capteur couvrant une zone ne peut déclencher qu'une alarme protégeant cette même zone.

Pour créer une contrainte dans le diagramme de classes :

1. Cliquez sur le bouton **Contrainte**  de la barre d'objets.



 *S'il n'est pas affiché, cliquez sur le menu **Affichage** > **Vues et détails** et cochez la case "Contraintes".*

2. Cliquez dans une des associations concernées par la contrainte, et faites glisser la souris jusqu'à la deuxième association, avant de relâcher votre pression.

La fenêtre d'ajout d'une contrainte s'ouvre.

3. Saisissez le nom de la contrainte puis cliquez sur **Ajouter**.

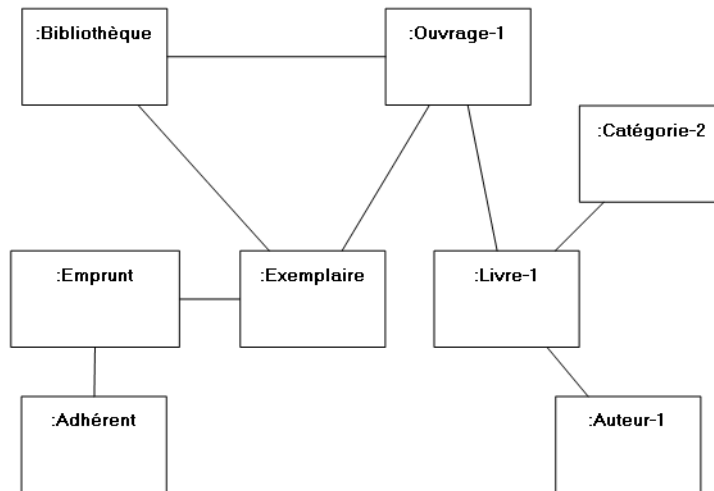
La contrainte apparaît dans le diagramme.

 *Vous pouvez relier une contrainte à d'autres classes ou associations à l'aide du bouton **Lien** .*

LE DIAGRAMME D'OBJETS

Un diagramme d'objets, ou diagramme d'instances, contient des objets avec des valeurs exemples pour leurs attributs, et des liens. Il montre en détail l'état du système à un instant précis.

Vous pouvez créer le diagramme d'objets d'une classe, d'un composant, d'un paquetage ou d'un cas d'utilisation.



Les objets

Un objet est une entité avec une identité et des frontières clairement définies dont l'état et le comportement sont encapsulés. Son état est défini par les valeurs de ses attributs et de ses liens avec d'autres objets. Son comportement est représenté par ses opérations et ses méthodes. Un Objet est une instance de Classe.


Exemples d'objets :

- Objets de gestion :
 - Jacques Dupond, Pierre Durand, Paul Smith sont des instances de la classe personne.
 - Les commandes no 10533 et 7322 sont des instances de la classe commande.
 - Ecran Sony SPD-1730, Compaq Deskpro 200 sont des instances de la classe article.
 - Dupond de Nemours, Burger King sont des instances de la classe société.
- Objets techniques utilisés pour la programmation :
 - Dlg_Order_Create, Dlg_Customer_Query sont des instances de la classe fenêtre.
 - Str_Customer_Name, Str_Product_Comment sont des instances de la classe chaîne.

☛ Les objets représentés dans un diagramme d'objets peuvent être des instances de classe, de paquetage, de cas d'utilisation, de composant ou de nœud, ce qui permet de définir des diagrammes de séquence au niveau de détail souhaité.

Créer un objet (une instance)

Pour créer un objet :

1. Cliquez sur le bouton **Instance** .
Vous pouvez créer des objets de différents types. La flèche située à droite du bouton offre un raccourci vers les types d'objets Classe ou Composant, plus fréquemment utilisés.
2. Puis cliquez sur le plan de travail du diagramme.
La fenêtre d'ajout d'une instance s'ouvre.
3. Saisissez le **Nom** de l'instance.
4. Précisez si nécessaire le **Type d'instance**.
5. Cliquez sur **Ajouter**.

L'instance est posée dans le diagramme.

Propriétés d'une instance

Pour ouvrir la fenêtre de propriétés d'une instance :

1. Sélectionnez l'instance en question et cliquez sur le bouton **Propriétés** de la fenêtre d'édition s'il n'est pas activé.

Plusieurs pages permettent de définir les propriétés d'une instance.

Sous la page **Caractéristiques**, vous pouvez :

- Sélectionner le **Type** de l'instance (Acteur, Classe, etc.).
- Préciser de quelle **Classe**, quel **Acteur**, etc. cet objet est une instance.
- Indiquer un nom pour cette Instance.
- Préciser son **Stéréotype**.

Valeur d'un attribut

Pour renseigner la valeur d'un attribut :

1. Affichez les propriétés de l'instance de la classe qui contient l'attribut.
2. Sélectionnez la page **Attributs**.
3. Dans la colonne correspondante, indiquez la valeur de l'attribut. Vous pouvez renseigner une valeur instanciée ou une valeur constante.
 - **Valeur instanciée** : cliquez dans cette colonne pour afficher la liste des instances possibles pour l'attribut sélectionné. Il s'agit de valeurs variables.
 - **Valeur** : cliquez dans la colonne et entrez la valeur de l'attribut.

Les liens


Un lien entre objets représente une instance d'association entre deux objets.

Exemples de liens entre objets :

- La commande n° 10733 a été passée par Jacques Dupond.
- La commande 10733 comprend les produits Ecran Sony SPD-1730 et Compaq Deskpro 200.
- Mr Jacques Dupond travaille pour la société Dupond de Nemours.
- La fenêtre Dlg_Customer_Query affiche la chaîne de caractères Str_Customer_Name.

Créer un lien

Pour créer un lien :

1. Cliquez sur le bouton **Lien**  de la barre d'outils du diagramme.
2. Cliquez sur l'un des objets concernés, et faites glisser la souris jusqu'au deuxième objet, avant de relâcher votre pression.


Le lien apparaît dans le dessin.

Si un lien entre les deux objets existe déjà, une fenêtre s'ouvre pour vous permettre de choisir parmi les liens existants ou en créer un nouveau.

Propriétés d'un lien

Pour ouvrir la fenêtre de propriétés d'un lien :

1. Sélectionnez le centre du lien pour afficher ses **Propriétés**.


 Si vous ne cliquez pas au centre du lien, c'est la fenêtre Propriétés d'un des rôles qui va s'ouvrir.

Sous la page **Caractéristiques**, vous pouvez préciser :

- Le **Nom** du lien.
- Le **Stéréotype** du lien.
- L'**Association** correspondant au lien.
- La **Visibilité** du lien.
- Le **Paquetage** détenteur du lien.

Et sous la page **Rôle du lien** :



- Pour chaque **Instance** reliée par ce lien, le nom du **Rôle** et la **Multiplicité** de ce rôle.

 Parmi les associations proposées ne figurent que celles qui figurent entre les classes des deux objets.

Propriétés d'un rôle

Pour ouvrir la fenêtre de propriétés d'un rôle :

1. Dans la fenêtre de propriétés d'un lien, sélectionnez la page **Rôle du lien**.
2. sélectionnez le rôle en question et cliquez sur **Propriétés**.

 Le bouton  permet d'afficher les commandes cachées.

La fenêtre de propriétés du rôle s'affiche.

Dans cette fenêtre vous pouvez préciser :

- Un **Nom** pour l'instance de rôle.
- Le **Rôle** de cette instance.
- La **Multiplicité** de l'instance de rôle.
- Pour cette instance de rôle, les valeurs des *qualificatifs* définis au niveau de la classe.

LES DIAGRAMMES DE STRUCTURE ET DE DÉPLOIEMENT



Outre les diagrammes de classes et d'objets, les diagrammes structurels comprennent :

- ✓ [Le diagramme de paquetages](#), qui permet d'organiser les éléments du modèle.
- ✓ [Le diagramme de composants](#), qui met en évidence les relations de dépendance entre composants.
- ✓ [Le diagramme de structure composite](#), qui décrit les interactions entre les composants et leurs parties.

LE DIAGRAMME DE PAQUETAGES

Un diagramme de paquetages permet d'organiser les éléments de modélisation, de manière à assurer une partition des travaux de spécifications et de développement.

Un élément ne doit apparaître que dans un seul paquetage.

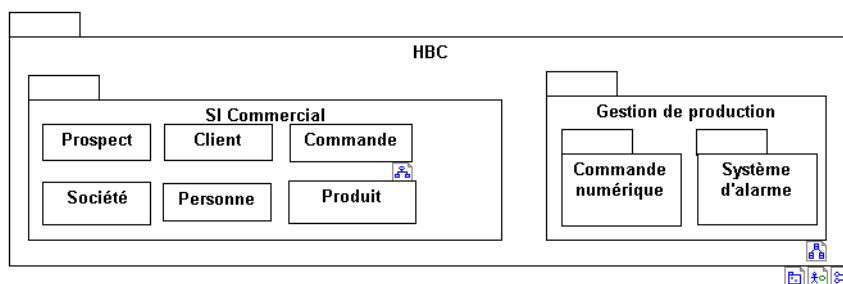
Le découpage en paquetage est généralement fait de manière à minimiser les interactions entre les différents paquetages.

Exemple de diagramme de paquetages

Le paquetage "HBC" contient les paquetages "SI Commercial" et "Gestion de production".

Le paquetage "Gestion de production" se décompose en deux paquetages "Commande numérique" et "Système d'alarme".

Le paquetage "SI Commercial" contient les classes "Prospect", "Client", "Société", "Personne", "Commande" et "Produit".



Créer un diagramme de paquetages

Un diagramme de paquetage se crée depuis un paquetage.

Pour créer un diagramme de paquetages dans **HOPEX Application Design** :

1. Cliquez sur le menu de navigation puis sur **Ressources des applications**.
2. Dans le volet de navigation, cliquez sur **Implémentation OO (UML)**.
3. Dans la fenêtre d'édition, cliquez sur la tuile **Paquetages**.
4. Affichez l'ensemble des paquetages.
5. Faites un clic droit sur le nom du paquetage concerné et sélectionnez **Nouveau > Diagramme de paquetages**.
Le diagramme créé s'ouvre dans la fenêtre d'édition.


Définir les paquetages

Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.

Exemples de *paquetages* :

- Le système d'information commercial.
- La comptabilité.
- La gestion de production.
- La commande numérique d'une machine.
- La gestion des stocks.
- La gestion du système d'alarme et du téléphone.

Pour ajouter dans le diagramme un paquetage existant :


1. Dans le diagramme de paquetages, cliquez sur le bouton **Paquetage** de la barre d'objets puis cliquez sur le plan de travail.
2. Dans la fenêtre **Ajout d'un paquetage**, sélectionnez la commande **Lister** à l'aide de la flèche .
La liste des paquetages apparaît.
3. Sélectionnez le paquetage qui vous intéresse et cliquez sur **OK**.
Le nom du diagramme apparaît dans la fenêtre **Ajout d'un paquetage**.
4. Cliquez sur **Ajouter**.

Le paquetage apparaît dans le diagramme.

Définir les classes

Le diagramme de paquetages permet de répartir les classes entre les paquetages.


Pour ajouter rapidement un ensemble de classes dans le diagramme de paquetages :

1. Cliquez sur **Menu principal** > **Recherche avancée** pour ouvrir l'assistant de recherche.
2. Dans l'assistant, sélectionnez la métaclasse "Classe" et cliquez sur **Chercher** .
La liste des classes du référentiel apparaît.
3. Sélectionnez les classes qui vous intéressent et glissez-les dans le diagramme.

Spécifier les dépendances dans un diagramme de paquetage


Des liens vous permettent d'indiquer si un paquetage détient ou référence une classe ou un autre paquetage.


Pour indiquer qu'un paquetage référence une classe ou un autre paquetage :

1. Cliquez sur le bouton 
2. Effectuez le lien en partant d'un paquetage vers le paquetage ou la classe qu'il référence.
Une fenêtre vous demande le type de lien à créer.
3. Sélectionnez "Paquetage référencé" ou "Classe référencée" selon qu'il s'agisse d'un paquetage ou d'une classe.

LE DIAGRAMME DE COMPOSANTS

Un diagramme de *composants* présente l'interdépendance des composants logiciels et des *interfaces* (il définit qui utilise quoi).

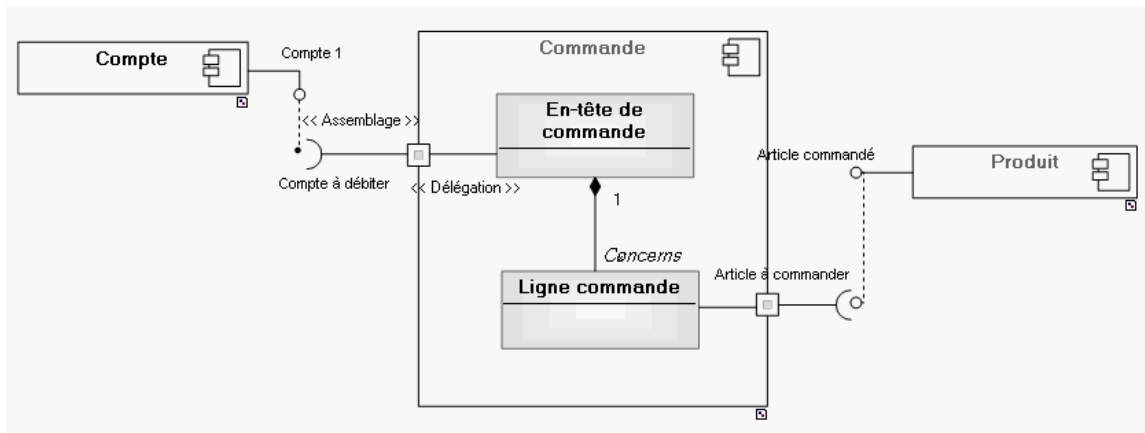
 *Un composant est un élément de la mise en oeuvre du système : ce peut être un logiciel, un programme, un élément de code, mais aussi un élément physique tel qu'un document de travail.*

 *Une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur. L'interface est un stéréotype de classe.*

Un diagramme de composants contient des composants et des classes de stéréotype interface. Il est également possible d'y préciser les paquetages implémentés par les composants.

Exemple de diagramme de composants

Ce diagramme décrit les éléments détenus par le composant "Commande" et les interactions de ces éléments avec des composants externes.



Créer un diagramme de composants

Dans **HOPEX Application Design** vous pouvez créer un diagramme de composants à partir d'un composant ou d'un paquetage.

Pour créer un diagramme de composants à partir d'un paquetage :

1. Cliquez sur le menu de navigation puis sur **Ressources des applications**.
2. Dans le volet de navigation, cliquez sur **Implémentation OO (UML)**.
3. Dans la fenêtre d'édition, cliquez sur la tuile **Paquetages**.
4. Affichez l'ensemble des paquetages.


5. Faites un clic droit sur le nom du paquetage concerné et sélectionnez **Nouveau > Diagramme de composants**.
Le diagramme apparaît dans la fenêtre d'édition.

Les composants

Un composant représente une partie modulaire d'un système qui encapsule son contenu et qui est remplaçable dans son environnement. Un composant définit son comportement par les interfaces qu'il fournit et celles qu'il requiert.

Un composant peut être remplacé par un autre si leurs interfaces sont conformes.

Un composant peut être un logiciel, un programme, un élément de code, etc.

Il est représenté par l'icône suivante : 


Les interfaces


Créer les interfaces des composants

Une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur.

L'interface est un type particulier de classe.

Pour créer une classe de stéréotype "Interface" dans le diagramme de structure composite :

1. Cliquez sur le bouton **Interface**  puis cliquez dans le diagramme.
2. Dans la fenêtre qui apparaît, saisissez le nom de la classe.
3. Cliquez sur **Ajouter**.

 Vous pouvez spécifier le détail de l'interface en termes d'attributs et d'opérations dans le diagramme de classes de la même manière que pour une classe.

Relier les interfaces aux autres objets

Deux types de lien permettent de différencier les interfaces requises des interfaces fournies.

Une interface requise est une interface nécessaire au fonctionnement de l'objet.


Exemple : le composant « Gestion des achats » a besoin pour son fonctionnement de l'interface « Produit » pour pouvoir associer une commande d'achat aux produits commandés.

Une interface fournie est une interface mise à disposition par un objet à destination d'autres objets.

Exemple : le composant « Gestion des produits » met à disposition l'interface « Produit ».

Vous pouvez définir les interfaces requises et les interfaces fournies par un objet indépendamment des autres objets.

Pour préciser qu'une interface est supportée ou requise par un objet :

1. Cliquez sur le bouton **Lien**  et tirez le lien en partant de l'objet client vers l'interface.
Une fenêtre apparaît.
2. Sélectionnez le type du lien à créer :
 - Interface requise
 - Interface supportée
3. Cliquez sur **OK**.

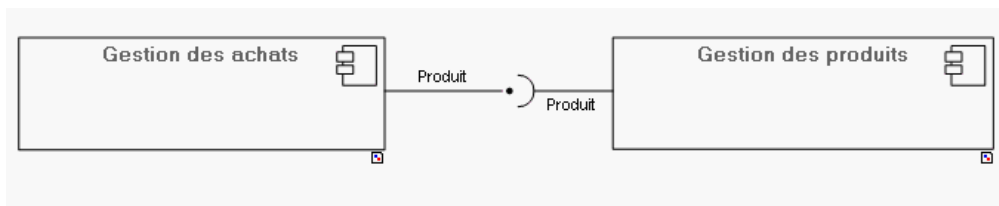
Le lien apparaît dans le dessin.

Selon le type de lien, la forme de l'interface change.

Relier des interfaces

Deux interfaces peuvent être reliées l'une à l'autre. Cette connexion est modélisée par un connecteur.

Vous pouvez également indiquer qu'une interface fournie par un objet est requise par un autre. Il s'agit ici d'une seule et même interface.



Les ports

Les ports permettent de connecter un composant à ses parties ou à son environnement.

Les ports sont symbolisés par un carré dans le diagramme, et posés en bordure du composant décrit lorsqu'ils assurent la connexion avec l'extérieur.

Ils sont reliés aux composants par des connecteurs.

Les ports peuvent spécifier les requêtes envoyées et les services fournis par le composant ainsi que les requêtes et services qu'ils peuvent requérir d'autres parties du système. Ces requêtes et services sont représentés par des classes de type Interface.

Vous pouvez visualiser les interfaces associées à un port dans la fenêtre de propriétés d'un port, sous l'onglet **Interfaces fournies et requises**.

Les connecteurs

Les connecteurs permettent de relier les objets du diagramme.

Les connecteurs de type simple ne spécifient aucun type de connexion particulier, ils sont utilisés notamment pour relier les instances d'objets décrits dans des collaborations.

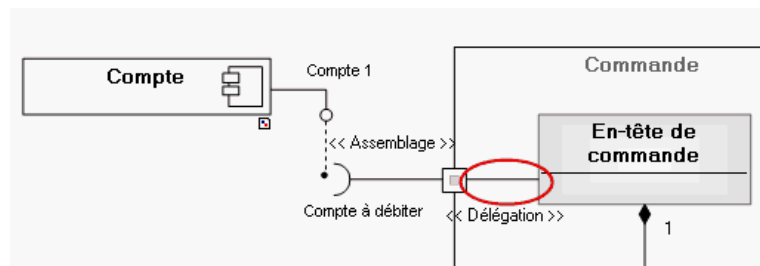
Dans le diagramme de structure composite, il est possible de spécifier le type de connecteur qui relie deux composants : Assemblage ou Délégation.

Connecteur de délégation

Un connecteur de type "Délégation" montre le réacheminement de requêtes vers un élément du composant chargé de les réaliser.

Le lien de délégation peut se faire directement entre le port du composant et l'élément du composant ou entre le port du composant et le port de l'élément.

Ci-dessous, le composant "Commande" délègue la gestion des comptes à débiter à la classe "En-tête de commande".



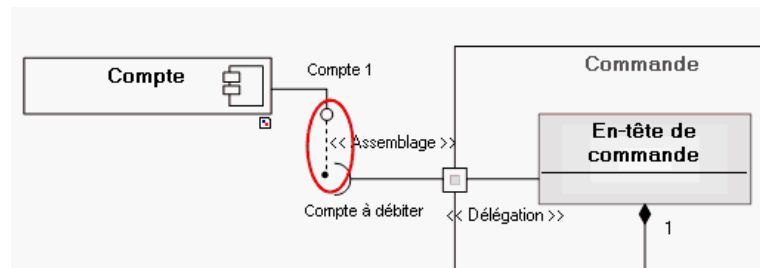
Connecteur d'assemblage

Un connecteur de type "Assemblage" est un connecteur entre deux ou plusieurs composants ou ports qui indique qu'un ou que plusieurs composants fournissent les services que d'autres utilisent.

☛ *Il peut s'agir d'autres objets que de composants.*

Pour relier des ports ou des composants qui partagent une interface, vous pouvez également utiliser les liens "Interface fournie" et "Interface requise".

Un connecteur de type "Assemblage" relie l'interface fournie par le composant "Compte" à l'interface requise par la classe "En-tête Commande".



LE DIAGRAMME DE STRUCTURE COMPOSITE

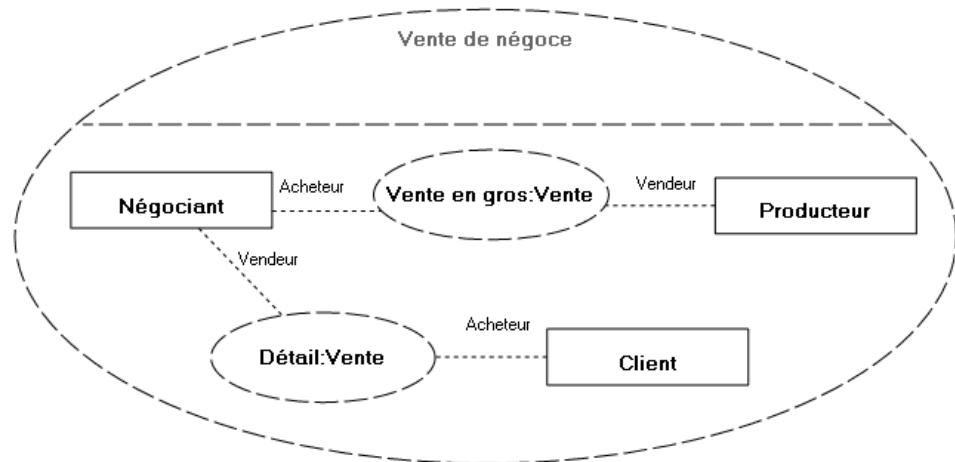
Le diagramme de structure composite permet de décrire la structure interne d'un composant, d'un paquetage ou d'une classe structurée.

Il permet également de préciser les collaborations qui interviennent entre les éléments de la structure dans l'exécution d'une tâche, en mettant en évidence le rôle joué par chaque élément dans la collaboration.

Les éléments de ce diagramme sont les parties (les *parts* en anglais), les ports par le biais desquels les parties interagissent avec l'extérieur, et les connecteurs reliant les parties entre elles ou avec les ports.

Exemple de diagramme de structure composite

Ce diagramme décrit le rôle joué par les parties dans la collaboration "Vente de négoce".



Créer un diagramme de structure composite

Pour créer un diagramme de composants à partir d'un composant :

1. Dans **HOPEX Application Design** cliquez sur le menu de navigation puis sur **Ressources des applications**.
2. Dans le volet de navigation, cliquez sur **Implémentation OO (UML)**.
3. Dans la fenêtre d'édition, cliquez sur la tuile **Composants**.
4. Faites un clic droit sur le nom du composant concerné et sélectionnez **Nouveau > Diagramme de composants**.

Le diagramme apparaît dans la fenêtre d'édition.

Les parties

Une partie représente un rôle joué par une instance d'une classe ou d'un composant lors de l'exécution d'une tâche.

Les parties sont reliées entre elles par des connecteurs ou des dépendances.

Une partie peut également être reliée - via un connecteur- à un port qui assure l'interface entre le composant décrit et l'extérieur.

Pour plus de détails sur ces éléments, voir :

- ✓ "Les connecteurs", page 210
- ✓ "Les liens de dépendance", page 214
- ✓ "Les ports", page 209.


Les multiplicités définies sur les parties indiquent le nombre d'instances qui sont créées. Les multiplicités sur les rôles de connecteur indiquent le nombre de liens qui peuvent être créés pour chacune de ces instances.

Pour définir la multiplicité d'une partie :

1. Ouvrez la fenêtre de propriétés de la partie.
2. Sélectionnez la page **Caractéristiques**.
3. Cliquez sur la flèche située à l'extrémité du champ **Multiplicité** et sélectionnez la multiplicité voulue.
4. Cliquez sur **OK**.


Les collaborations

Dans le diagramme de structure composite, une *collaboration* décrit les rôles joués par chaque partie (instance) dans la réalisation d'une tâche.

 Une collaboration (UML) décrit une structure collaborative entre plusieurs éléments (rôles), qui accomplissent chacun une fonction spécialisée et qui réalisent collectivement une fonctionnalité attendue du système. Son objectif est de montrer comment un système fonctionne indépendamment d'une utilisation spécifique. On en retirera donc généralement l'identité précise des classes ou des instances qui y participent.

Elle est représentée par un ovale en pointillé contenant les instances de la collaboration.

Ces instances sont reliées entre elles par des *connecteurs*. A chaque extrémité du connecteur s'affiche le rôle qui correspond au nom de l'instance.

 Un connecteur est un lien qui permet d'établir une communication entre plusieurs objets. Un connecteur de délégation relie le contrat externe de l'objet (tel qu'il est spécifié par ses ports et/ou ses interfaces) aux objets internes qui vont le réaliser. Un connecteur d'assemblage entre plusieurs objets (ou leurs ports) permet de spécifier comment un des objets fournit l'interface requise par un autre.

Il est possible d'appliquer le modèle d'une collaboration à différentes instances.

Utilisation de collaboration

Une utilisation de collaboration représente l'application de la structure décrite par une collaboration à une situation particulière mettant en oeuvre des classes ou des instances spécifiques. Ces classes ou ces instances jouent alors les rôles définis dans la collaboration.

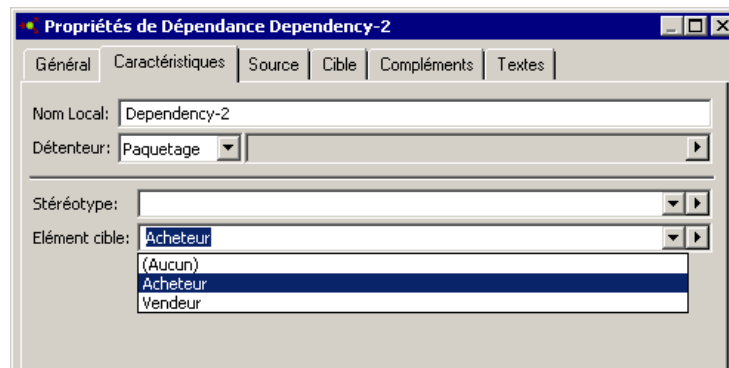
Les instances sont reliées à l'utilisation d'une collaboration par un lien de *dépendance* sur lequel doit être précisé le rôle joué par l'instance.



Une dépendance précise que l'implémentation ou le fonctionnement d'un ou de plusieurs éléments nécessite la présence d'un ou de plusieurs autres éléments. Il existe plusieurs stéréotypes de dépendance.

Exemple d'utilisation de collaboration

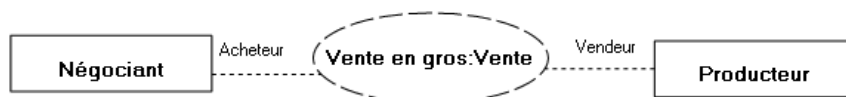
Dans le cas d'une demande d'achat entre deux instances d'acteur, une collaboration est utilisée. Cette collaboration relie deux rôles : le rôle d'acheteur et le rôle de vendeur. Sur la dépendance qui relie chaque instance à la collaboration, vous pouvez indiquer le rôle que joue l'instance.



Les liens de dépendance

Une dépendance précise que l'implémentation ou le fonctionnement d'un ou de plusieurs éléments nécessite la présence d'un ou de plusieurs autres éléments.

Une dépendance est également une relation de type fournisseur / client qui indique quel est l'élément source et l'élément cible dans la collaboration.



Un stéréotype sur la dépendance permet de spécifier la nature de la dépendance :

- Binding : le binding est une relation entre un template et un élément de modélisation généré à partir du template. Il inclut une liste d'arguments en correspondance avec les paramètres du template.
- Derive : indique une relation de dérivation entre des éléments de modélisation qui sont généralement, mais pas nécessairement, de même type. Une telle relation de dépendance implique que l'un des éléments peut être calculé à partir de l'autre.
- Mapping UML/XML : une expression de mapping qui définit la relation entre les éléments (classes, attributs, ...) d'un schéma ou d'un diagramme de classes et ceux d'un autre schéma ou diagramme de classes.
- Refine : spécifie une relation de dépendance entre des éléments de modélisation à différents niveaux sémantiques, tels que l'analyse et la conception.
- Trace : spécifie une relation de traçabilité entre des éléments de modélisation ou des ensembles d'éléments de modélisation qui représentent le même concept dans différents modèles.

Pour préciser la nature d'une dépendance :

1. Ouvrez la fenêtre de propriétés de la dépendance.
2. Sélectionnez la page **Caractéristiques**.
3. Dans le champ **Stéréotype**, déroulez la liste et sélectionnez l'un des stéréotypes proposés.

La flèche  vous permet également de créer de nouveaux stéréotypes.

LE DIAGRAMME DE MACHINE À ÉTATS



Un diagramme de machine à état permet de décrire les comportements possibles d'un objet, suivant les événements auxquels il est soumis au cours de son cycle de vie.

Les points suivants sont abordés ici :

- ✓ ["Présentation du diagramme de machine à états", page 218](#)
- ✓ ["Créer un diagramme de machine à états", page 218](#)
- ✓ ["Les états", page 220](#)
- ✓ ["Les transitions entre états", page 224](#)

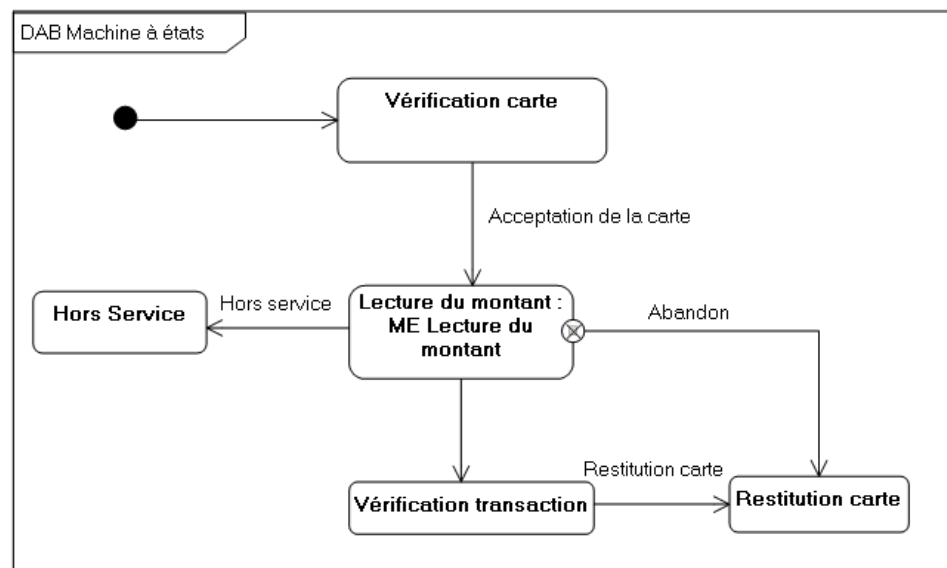
PRÉSENTATION DU DIAGRAMME DE MACHINE À ÉTATS

Une machine à état est l'ensemble des états et des transitions entre états qui définissent le cycle de vie d'un objet variable dans le temps.

Le diagramme de machine à état permet de représenter cet enchaînement d'états que peut prendre un objet en réponse aux interactions avec les objets (internes ou externes au système étudié) qui peuplent son environnement.

Exemple de diagramme de machine à d'états

Le diagramme suivant décrit les comportements possibles d'un distributeur automatique.



Créer un diagramme de machine à états

Un diagramme de machine à états se crée à partir d'une machine à état.

Vous pouvez créer une machine à état à partir d'un paquetage, d'une classe ou d'un composant.

Pour créer un diagramme de machine à états dans **HOPEX Application Design** :

1. Cliquez sur le menu de navigation puis sur **Ressources des applications**.
2. Dans le volet de navigation, cliquez sur **Implémentation OO (UML)**.
3. Dans la fenêtre d'édition, cliquez sur la tuile **Machines à états**.

4. Faites un clic droit sur le nom de la machine à états concernée et sélectionnez **Nouveau > Diagramme de machine à états**.
Le diagramme créé s'ouvre dans la fenêtre d'édition.

Le diagramme est initialisé par la création d'une région. Une région est une partie d'un état composite ou d'une machine à états qui contient des états et des transitions et dont l'exécution est autonome.

LES ÉTATS


Un état d'objet est une condition ou une situation au cours de la vie d'un objet durant laquelle il satisfait à certaines conditions, exerce une certaine activité ou attend un événement. Un état d'objet représente un intervalle de temps dont les bornes sont deux événements. Un état d'objet est une phase par laquelle passe l'objet au cours de son cycle de vie.

Exemples d'état d'objet

- Une personne peut être :
 - Célibataire
 - Mariée
 - Divorcée
- Un article peut être :
 - Disponible
 - En stock
 - En alerte
 - En rupture de stock.
 - Etc.





Créer un état

Pour créer un état dans le diagramme de machine à états :

1. Cliquez sur la flèche noire associée au bouton **Etat**  de la barre d'insertion du diagramme.
2. Sélectionnez un type d'état.
3. Cliquez sur le plan de travail.
La fenêtre **Ajout d'un état** s'ouvre.
4. Indiquez le **Nom** de l'état et cliquez sur **Créer**.
L'état apparaît dans le diagramme.

Les types d'état

Il est nécessaire de préciser le type de l'état lors de sa création. Ce peut être :

-  Un état normal : ne possède pas de sous-structure.
-  Un état composite : se compose de plusieurs états, décrits dans le diagramme.
-  Une sous-machine à état : appelle la description d'une machine à état décrite par ailleurs. Voir "[Précision comportementale d'un état](#)", [page 222](#).
-  Un état final

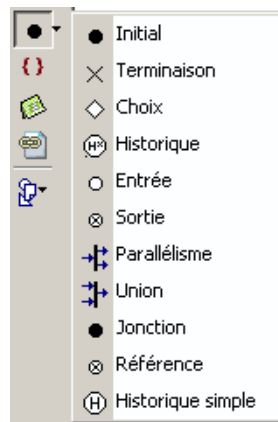
Lorsque vous posez un état dans un autre, il est automatiquement relié comme composant de cet état.

Pseudo-états

Les pseudo-états sont utilisés pour spécifier des chemins complexes en combinant plusieurs transitions entre états.

Ils peuvent être de différents types : initial, terminaison, choix, historique (deep history), historique simple (shallow history), point d'entrée, point de sortie, parallélisme (fork), union (join), jonction ou référence.

Initial



Le pseudo-état initial a une seule transition en sortie vers l'état Initial de l'objet lors de sa création.

Historique

Un pseudo-état Historique représente la dernière configuration active de l'état composite qui le contient ; c'est-à-dire, la configuration active quand l'état composite a été quitté pour la dernière fois.

Historique simple

Un pseudo-état historique simple représente le plus récent sous-état actif d'un état composite (sans les sous-états de ce sous-état).

Parallélisme

Un parallélisme (fork) sépare une transition en plusieurs transitions concurrentes.

Union

Une union (join) est le regroupement de plusieurs transitions en une seule.

Choix

Représente le choix d'une transition entre plusieurs transitions possibles.

Jonction

Une jonction est utilisée pour définir des chemins de transition complexes entre plusieurs états.

Entrée

C'est un point d'entrée d'une machine à état ou d'un état composite.

Sortie

C'est un point de sortie d'une machine à état ou d'un état composite.

Référence

C'est une référence à une entrée ou à une sortie d'une machine à état ou d'un état composite

Terminaison

L'entrée dans ce pseudo-état implique une terminaison complète de la machine à état.

Historique

Un état **Historique** représente la dernière configuration active d'un état composite ; c'est-à-dire la configuration active quand l'état composite a été quitté pour la dernière fois.

Un état **Historique simple** représente le plus récent sous-état actif de l'état composite.

Exemple :

Prenons l'état "Marié" comme dernière configuration active. Cet état a pour sous-états "Avec enfants" et "Sans enfants". Dans le cas d'un historique, le sous-état "Avec enfants" ou "Sans enfant" est précisé. Dans le cas d'un historique simple, seul l'état "Marié" est pris en compte.

Précision comportementale d'un état

Un état peut être composé de sous-états.

Pour décrire la composition d'un état dans un diagramme :

1. Ouvrez le menu contextuel d'un état et cliquez sur la commande **Nouveau > Précision comportementale**.
La fenêtre de création d'un diagramme de machine à états apparaît.
2. Cliquez sur **Créer**.
Le diagramme correspondant s'ouvre.

Vous pouvez également définir la composition d'un état en lui associant une machine à état, nouvelle ou existante :

1. Ouvrez la fenêtre de propriétés de l'état décrit.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Dans le champ **Précision comportementale**, créez une machine à état ou recherchez une machine à état existante.

Propriétés d'un état

Pour accéder aux propriétés d'un état :

1. Cliquez avec le bouton droit sur l'état.
2. Sélectionnez **Propriétés**.
La fenêtre de propriétés de l'état s'ouvre.

Elle vous permet :

- De modifier le **Nom** de l'état.
- D'indiquer si ses sous-états sont **Concurrents**, c'est-à-dire s'ils peuvent être exécutés simultanément ou non.
- D'indiquer la **Précision comportementale** (dans le cas d'un état complexe). Voir "[Précision comportementale d'un état](#)", page 222.
- De préciser les **Activités** qui peuvent être effectuées en entrée, en sortie ou pendant que l'objet est dans cet état.

☞ Le contenu de la fenêtre de propriétés d'un état varie en fonction du type de l'état.

LES TRANSITIONS ENTRE ÉTATS

Le passage d'un état à un autre est matérialisé par une **transition**.



*Une transition est le passage d'un objet d'un état dans un autre.
Une transition est une réponse d'un objet à un événement qu'il reçoit.
Quand un événement se produit et que certaines conditions sont satisfaites, l'objet va effectuer certaines actions tandis qu'il est encore dans le premier état puis passer au deuxième état.*

Vous devez définir toutes les transitions qui sont autorisées. Celles qui ne sont pas définies sont interdites.

Exemples de transitions :

En ce qui concerne l'état civil d'une personne, certaines transitions sont possibles :

- Elle peut passer de l'état "célibataire" à l'état "marié".
- Elle peut passer de l'état "marié" à l'état "divorcé".

D'autres ne sont pas possibles :

- Elle ne peut pas passer de l'état "célibataire" à l'état "divorcé".

Créer une transition

Pour créer une transition entre deux états :

1. Dans le diagramme de machine à états, cliquez sur le bouton **Transition**

(UML)



de la barre d'insertion.

2. Cliquez sur l'état de départ et déplacez la souris jusqu'à l'état d'arrivée.
3. Relâchez le bouton : la transition est créée.

Les types de transition

Une transition peut être de type externe, interne ou locale.

Vous pouvez préciser le type de la transition dans la fenêtre de propriétés de la transition, sous l'onglet **Caractéristiques**.

Transition externe

Une transition externe est une transition qui modifie l'état actif.

Transition interne

Une transition interne à un objet permet de prendre en compte l'arrivée d'un événement qui ne provoque pas de changement d'état de l'objet, mais une action comme l'appel d'une opération ou l'émission d'un message. Par exemple, lors d'un

mouvement de stock, un article peut ne pas changer d'état si la quantité restant disponible en stock est suffisante et ne passe pas le seuil d'alerte ou de rupture.

Transition locale

Une transition locale s'applique aux sous-états d'un état composite. Elle peut provoquer un changement d'état uniquement à l'intérieur de l'état composite.

Effet d'une transition

Le déclenchement d'une transition peut être accompagné d'un effet. L'effet peut être représenté par :

- Une activité
- Une collaboration
- Une interaction
- Une machine à état

Pour définir l'effet d'une transition :

1. Ouvrez la fenêtre de propriétés de la transition.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Cliquez sur la flèche située à l'extrémité du champ **Effet (Comportement)** et créez ou reliez l'objet qui définit l'effet.

Affichage des effets d'une transition

Pour modifier l'affichage des effets de la transition.

1. Dans le diagramme de machine à état, cliquez avec le bouton droit sur la transition puis cliquez sur **Formes et détails**.
2. Sélectionnez "Effet" dans l'arbre qui s'affiche.

Vous pouvez choisir d'afficher tout ou partie des effets de la transition, avec leurs caractéristiques.

Événement déclencheur d'une transition

Dans la fenêtre de propriétés d'une transition, sous l'onglet **Événement**, vous pouvez indiquer le **Type d'Événement** qui déclenche une transition.

Ce peut être :

- Un événement quelconque
- L'appel d'une opération
- Un changement de l'objet concerné par la transition
- La création d'un objet
- La destruction d'un objet
- L'envoi d'un signal
- L'envoi d'une opération
- L'émission d'un signal par l'objet
- La réception d'un signal
- La réception d'une opération
- Un *temporisateur*



Un temporisateur est un événement déterminé uniquement par le temps qui s'écoule. Ex : Le lundi, à quatre heures, etc.

Les champs affichés sous le champ **Type d'événement** varient selon le type d'événement sélectionné.

Vous pouvez sélectionner l'objet concerné par l'effet.

Dans le cas d'une opération ou d'un signal, il est possible de préciser les valeurs des paramètres transmis.

LE DIAGRAMME D'ACTIVITÉS



Un diagramme d'activités est proche du diagramme de machine à états. A la différence du diagramme de machine à états qui décrit le comportement d'un objet via l'enchaînement d'états, le diagramme d'activités décrit le comportement d'un élément en termes d'actions.

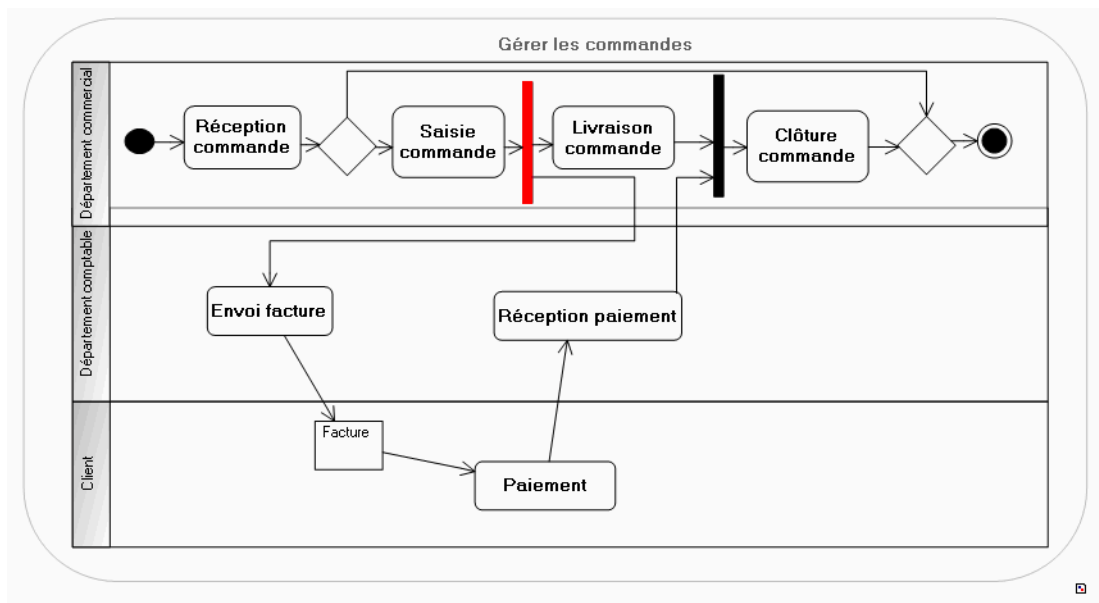
- ✓ ["Présentation du diagramme d'activités", page 228](#)
- ✓ ["Les partitions", page 229](#)
- ✓ ["Les noeuds", page 231](#)
- ✓ ["Les Flux", page 233](#)

PRÉSENTATION DU DIAGRAMME D'ACTIVITÉS

Un diagramme d'activités représente un séquençement d'étapes décrivant le comportement d'un élément du système.

Les étapes sont modélisées par des noeuds - noeuds d'action, de paramétrage ou de contrôle - coordonnés par des flux de données ou de contrôle.

Exemple de diagramme d'activités



Créer un diagramme d'activités

Dans **HOPEX Application Design**, un diagramme d'activités se crée à partir d'un paquetage ou d'une activité.

Vous pouvez créer une activité sur un paquetage, un composant ou une classe.

Pour créer un diagramme d'activités :

1. Faites un clic droit sur le paquetage ou l'activité concerné(e).
2. Dans le menu contextuel qui s'affiche, cliquez sur **Nouveau > Diagramme d'activités**.

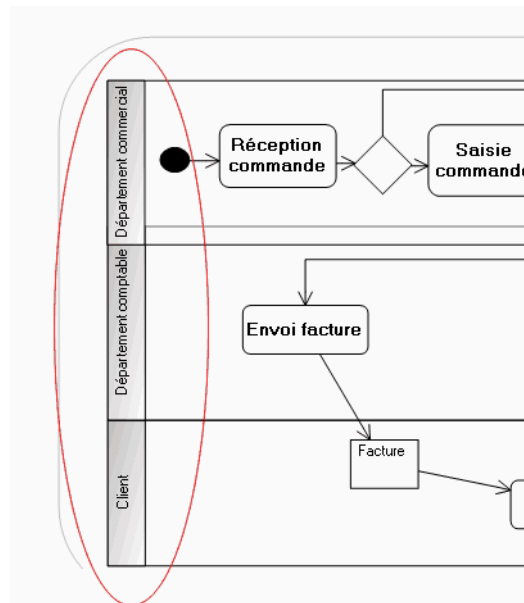
Le nouveau diagramme d'activités s'ouvre.

LES PARTITIONS

Un diagramme d'activités peut être découpé en partitions. Chaque partition contient des nœuds ou des actions ainsi que les flux entre ces éléments.

Vous pouvez utiliser des partitions pour organiser les tâches ou pour spécifier l'élément responsable de la mise en œuvre d'un ensemble de tâches.

Pour plus de détails sur les couloirs, voir le guide **HOPEX Common Features**, chapitre "Manipuler les diagrammes", section "Utiliser les couloirs".



Créer une partition

Pour créer une partition dans le diagramme d'activités :

1. Cliquez sur le bouton **Partition**  de la barre d'insertion d'objets.
2. Indiquez son nom.
3. Cliquez sur **Ajouter**.

Propriétés d'une partition

La page **Etats** présente les états contenus dans la partition.

La page **Compléments** permet d'indiquer l'élément représenté par la partition. Il s'agit de l'élément qui met en œuvre les éléments de la partition. Ce peut être un acteur, une classe ou un composant.

LES NOEUDS

Les noeuds permettent de modéliser les étapes de l'activité. Il existe différents types de noeuds dans **HOPEX**.

- "Les noeuds d'actions", page 231
- "Les noeuds de paramétrage", page 231
- "Les noeuds de contrôle", page 232
- "Les noeuds d'objets : pins d'entrée, de sortie et d'échange", page 233


Les noeuds d'actions

Les actions sont les étapes élémentaires du comportement représenté par l'activité.

La coordination des actions est réalisée à l'aide de flux de contrôle et de flux de données.

Créer une action

Pour créer une action dans un diagramme d'activités :

1. Dans la barre d'insertion d'objets du diagramme, sélectionnez le bouton correspondant au type d'action, puis cliquez sur le plan de travail.
La fenêtre d'ajout d'une action du type choisi s'ouvre.
 La barre d'insertion propose les trois types d'action principaux.
2. Indiquez son nom et cliquez sur **Ajouter**.

Modifier le type d'action

Dans la fenêtre de propriétés de l'action, sous la page **Caractéristiques**, vous pouvez modifier le type d'action. Ce peut être :

- L'appel d'une opération d'un autre objet
- La création d'un objet
- La destruction d'un objet
- L'exécution d'une opération locale à l'objet
- L'émission d'un signal par l'objet
- La destruction finale de l'objet
- Etc.

Les noeuds de paramétrage

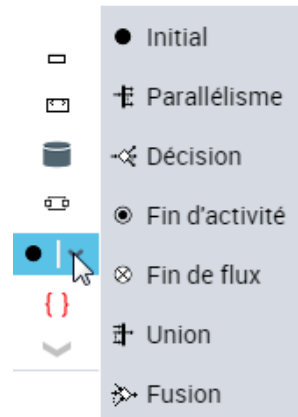
Les noeuds de paramétrage d'une activité décrivent les entrées ou les sorties de cette activité.

Ils transmettent les paramètres à l'activité par l'intermédiaire des flux qu'ils émettent ou qu'ils reçoivent.

Les nœuds de contrôle

Un nœud de contrôle coordonne les flux entre les noeuds d'une activité.

Un nœud de contrôle peut être de type initial, final, décision, fusion (merge), parallélisme (fork) ou union (join).



Types de noeud de contrôle

Initial

Un noeud initial indique où débute le flux de contrôle lorsque l'activité est invoquée. Une activité peut avoir plusieurs noeuds initiaux.

Final

Lorsqu'un jeton atteint un noeud final d'activité, tous les flux de l'activité sont stoppés. Au contraire, un noeud final de flux détruit les jetons qui lui arrivent mais n'a aucun effet sur les autres jetons de l'activité.

Décision

Une décision fait le choix d'un seul flux entre plusieurs flux sortants possibles. Les flux sortants sont sélectionnés en fonction de leur condition de garde.

Fusion

Une fusion (merge) rassemble plusieurs flux alternatifs entrants en un seul flux sortant. Elle n'est pas utilisée pour synchroniser des flux concurrents mais pour accepter un seul flux parmi plusieurs.

Parallélisme

Un parallélisme (fork) sépare un flux en plusieurs flux concurrents. Les jetons arrivant à un parallélisme sont dupliqués à travers les flux sortants.

Union

Une union (join) synchronise des flux multiples. Quand tous les flux en entrée sont disponibles, le flux en sortie est déclenché.

Les nœuds d'objets : pins d'entrée, de sortie et d'échange

Pour spécifier les valeurs en entrée d'une action et les valeurs de retour, on utilise des nœuds d'objets appelés pins (pin en anglais) d'entrée ou de sortie. L'action ne peut débuter que si une valeur est affectée au pin d'entrée. De même, quand l'action se termine, une valeur doit être affectée au pin de sortie.

Pin d'entrée

Un pin d'entrée supporte les valeurs d'entrée qui doivent être consommées par une action et qu'il reçoit de la part d'autres actions.

Pin de sortie

Un pin de sortie supporte les valeurs de sortie qui sont produites par une action et fournit ces valeurs à d'autres actions à travers des flux.

Pin d'échange

Un pin d'échange est utilisé pour représenter les données échangées entre deux actions.

Les Flux

Le passage d'un nœud à un autre est matérialisé par un flux.

Flux de contrôle

Un flux de contrôle démarre un nœud d'action lorsque le précédent est terminé. Les objets et les données ne peuvent pas être transmis par un flux de contrôle.

Flux d'objets

Un flux d'objets permet de transmettre des données ou objets d'un nœud à un autre à l'intérieur d'une activité.

LES DIAGRAMMES D'INTERACTION



Les diagrammes d'interaction, c'est-à-dire le diagramme de séquence, le diagramme de communication et le diagramme de vue générale d'interaction, représentent une série d'interactions entre objets, ordonnée dans le temps. Ils montrent une ou plusieurs histoires possibles du système.

Les points suivants sont abordés ici :

- ✓ ["Les interactions", page 236](#)
- ✓ ["Le diagramme de séquence", page 237](#)
- ✓ ["Le diagramme de communication", page 248](#)
- ✓ ["Le diagramme de vue générale d'interaction", page 250](#)

LES INTERACTIONS

Une interaction décrit le comportement d'un système dans un contexte particulier par les échanges de messages entre les éléments de ce système.

Quand les diagrammes de machines à état ou d'activités étudient des comportements individuels, les diagrammes d'interaction se concentrent sur la coopération d'un ensemble d'objets.

Créer une interaction

Vous pouvez créer une interaction à partir d'un composant, d'un paquetage ou d'une classe.

Pour créer une interaction dans **HOPEX Application Design** :

1. Cliquez sur le menu de navigation puis sur **Ressources des applications**.
2. Dans le volet de navigation, cliquez sur **Implémentation OO (UML)**.
3. Dans la fenêtre d'édition, cliquez sur la tuile **Interactions**.
4. Cliquez sur le bouton **Nouveau**.
5. Indiquez le nom de l'interaction et éventuellement le détenteur.
6. Cliquez sur **OK**.

Créer un diagramme d'interaction

Le diagramme de séquence, le diagramme de communication et le diagramme de vue générale d'interaction se crée à partir d'une interaction.

Pour créer un diagramme d'interaction :

1. Faites un clic droit sur une interaction.
2. Dans le menu contextuel qui s'affiche, cliquez sur **Nouveau > Diagramme d'interaction**.

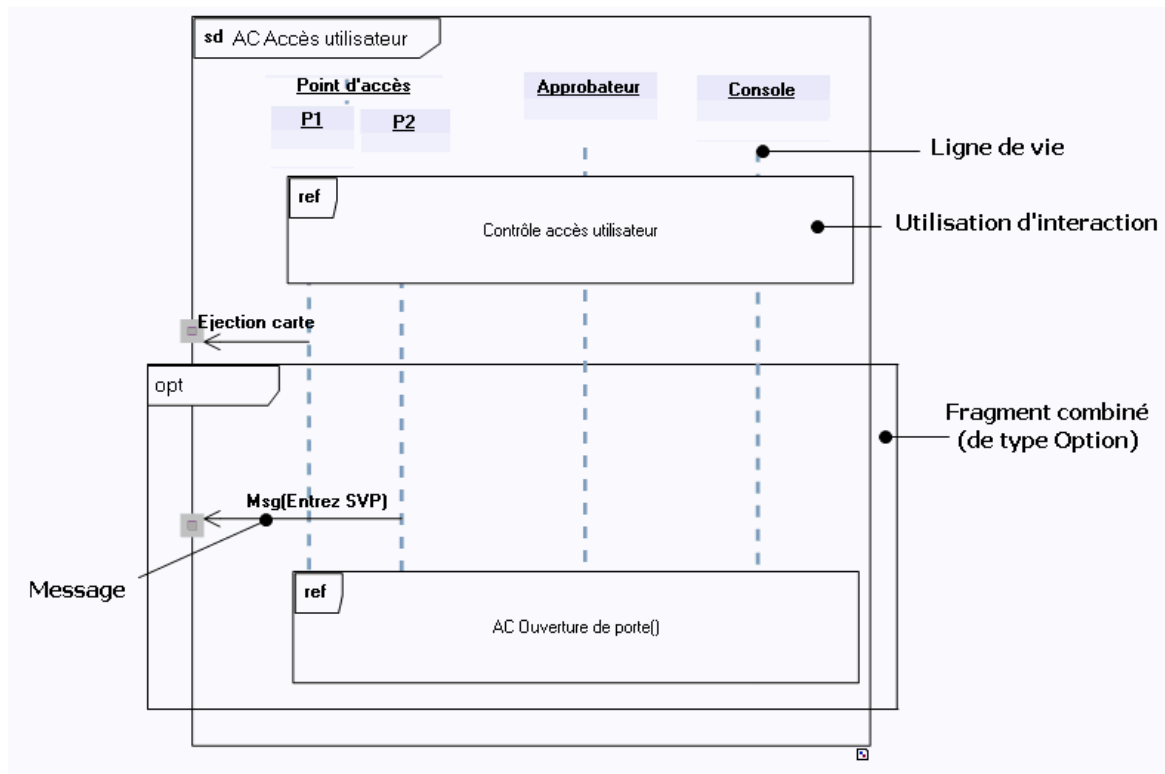
LE DIAGRAMME DE SÉQUENCE

Le diagramme de séquence met en évidence la chronologie des messages échangés entre les objets participant à une interaction. Ces objets sont représentés dans le diagramme par leurs lignes de vie.

Exemple de diagramme de séquence

Le diagramme ci-dessous décrit le comportement d'un distributeur automatique :

- Deux points d'entrée (représentés par des lignes de vie) donnent lieu à un contrôle d'accès de l'utilisateur. Ce contrôle est décrit dans une interaction.
- Selon le résultat du contrôle, l'accès est refusé et la carte de l'utilisateur éjectée ou l'ouverture de la porte est actionnée.
- Un comportement optionnel (représenté par un fragment combiné) peut influencer l'ouverture de la porte.



Créer un diagramme de séquence

Pour créer un diagramme de séquence dans **HOPEX Application Design** :

1. Faites un clic droit sur une interaction.
2. Dans le menu contextuel qui s'affiche, cliquez sur **Nouveau > Diagramme d'interaction**.


Voir aussi ["Créer une interaction"](#), page 236.

Les lignes de vie

Une ligne de vie représente un participant dans une interaction.


Les lignes de vie sont des instances de différents types (de classe, d'acteurs, etc.).

Dans un diagramme de séquence, le temps est représenté comme s'écoulant du haut vers le bas le long des lignes de vie de ces objets. Entre ces objets transitent des instances de messages.

 Les instances représentées dans un diagramme de séquence peuvent être des instances de classe, d'acteur, de paquetage, de cas d'utilisation, de composant ou de nœud, ce qui permet de définir des diagrammes de séquence au niveau de détail souhaité.

Créer une ligne de vie

Pour créer une ligne de vie :

1. Cliquez sur le bouton **Ligne de vie** .
2. Cliquez dans le diagramme.
Une fenêtre s'ouvre.
3. Saisissez le nom de la ligne de vie.
4. Cliquez sur **Ajouter**.
La ligne de vie apparaît dans le diagramme.

Propriétés d'une ligne de vie

Pour accéder aux propriétés d'une ligne de vie :

1. Sélectionnez l'instance et cliquez sur le bouton **Propriétés** de la fenêtre d'édition s'il n'est pas activé.

Vous pouvez sélectionner le **Type** de l'objet (Acteur, Classe, etc.), préciser de quelle **Classe**, **Acteur**, etc. cet objet est une instance et indiquer son *stéréotype*.

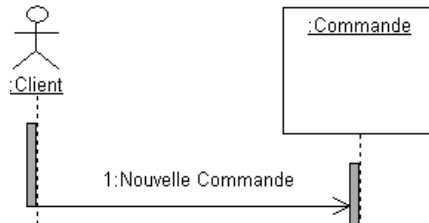
Les messages

Un message définit une communication particulière entre les lignes de vie d'une interaction. Il spécifie l'émetteur et le récepteur par l'intermédiaire de spécifications d'occurrence, ainsi que le type de communication. Cette communication peut être,

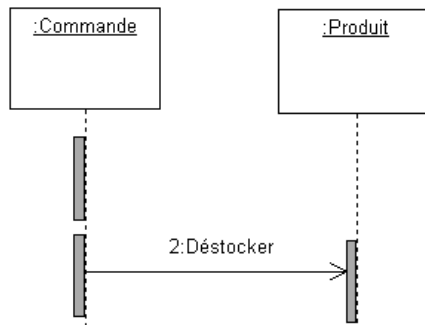
par exemple, l'émission d'un signal, l'invocation d'une opération, la création ou la destruction d'une instance.

Exemples de messages échangés

1) Le message envoyé par l'acteur "Client" à la classe "Commande" transporte le signal "Nouvelle commande".



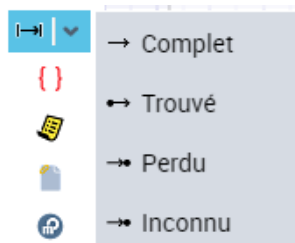
2) Le message envoyé par la classe "Commande" à la classe "Produit" appelle l'opération "Déstocker".



Créer un message

Pour créer un message dans le diagramme de séquence :

1. Cliquez sur le bouton **Message** de la barre d'insertion d'objets en sélectionnant le type de message voulu.



2. Allez de la ligne pointillée sous le premier objet à celle qui est sous le deuxième objet en maintenant le bouton gauche de la souris enfoncé. Le message échangé entre les deux objets se dessine.

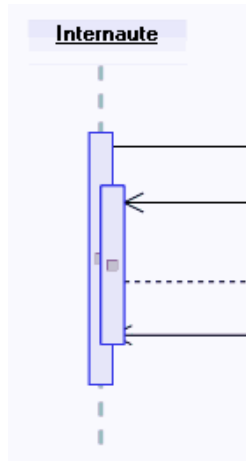
Types de messages

Vous pouvez créer quatre types de messages :

- Dans un message de type "Complet", l'émetteur et le destinataire sont tous les deux définis.
- Dans un message "Perdu", seul l'émetteur est connu. On considère ici que le message n'atteint jamais sa destination.
- Dans un message "Trouvé", seul le destinataire est connu. C'est le cas lorsque l'origine du message se situe en dehors du contexte de description.
- Dans un message de type "Inconnu", ni l'émetteur ni le destinataire ne sont définis.


Occurrence d'exécution

Une occurrence d'exécution d'une ligne de vie (execution specification) représente une unité d'action ou de comportement qui se déroule à partir d'une occurrence d'événement de début jusqu'à une occurrence d'événement de fin.



Créer une occurrence d'exécution

Pour créer une occurrence d'exécution :

1. Dans le diagramme de séquence, cliquez sur le bouton **Occurrence d'exécution**  de la barre d'insertion d'objets.
2. Positionnez-la sur la ligne de vie concernée. L'occurrence apparaît dans le diagramme.

Occurrence d'événement

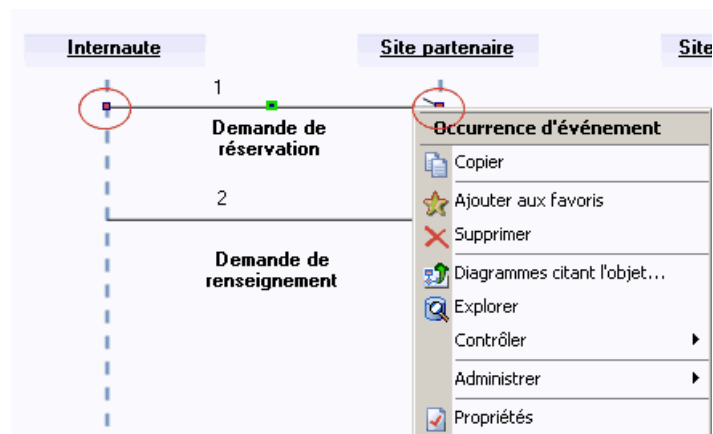
La création d'un message ou d'une occurrence d'exécution entraîne automatiquement la création d'occurrences d'événements.

Une occurrence d'événement (Occurrence Specification) est un point syntaxique à l'extrémité d'un message ou au début ou à la fin d'une occurrence d'exécution.

Les occurrences d'événement sont ordonnées le long d'une ligne de vie.

Ce sont les unités sémantiques de base d'une interaction.

Vous pouvez accéder au menu contextuel d'une occurrence d'événement en cliquant avec le bouton droit sur l'une des extrémités d'un message.



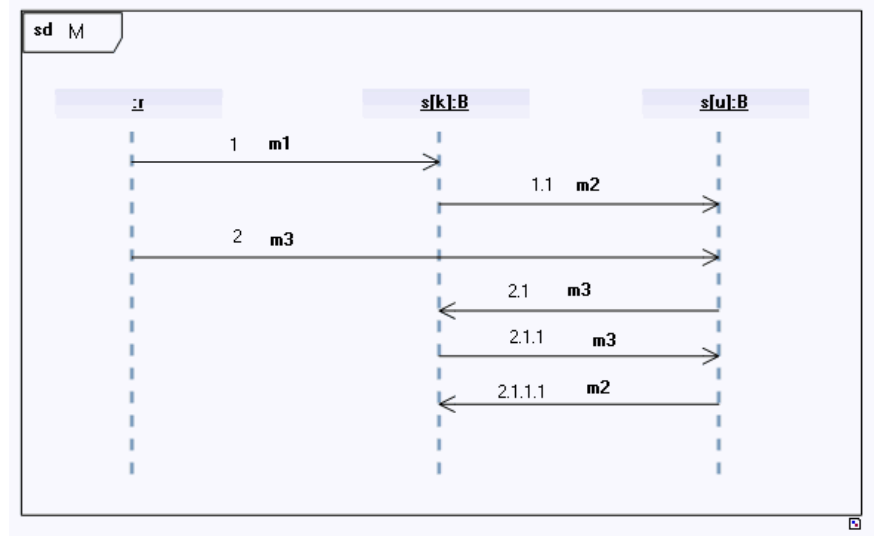
Calcul des numéros de séquence

A partir du positionnement des occurrences d'événement, un outil de calcul permet d'ordonner les messages et les occurrences d'exécution.

Pour ordonner les messages circulant entre des lignes de vie d'une interaction :

1. Ouvrez le menu contextuel de l'interaction décrite.
2. Cliquez sur **Calculer les numéros de séquence**.
L'outil applique automatiquement des numéros aux messages.

Exemple



Vous pouvez modifier manuellement le numéro de séquence d'un message dans la fenêtre de propriétés du message :

- 1 Cliquez sur l'onglet **Caractéristiques** et modifiez la valeur du champ **Sequence Expression**.

Lorsque vous relancez le calcul des numéros de séquence, ce dernier met à jour le séquençement en fonction des modifications apportées.

Fragment combiné

Un fragment combiné permet de décrire de manière concise plusieurs séquences d'exécution.

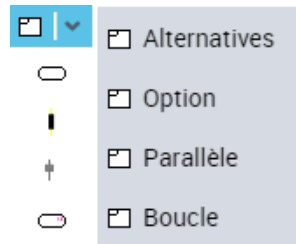
Un fragment combiné est défini par un opérateur d'interaction et les opérandes d'interactions correspondants.

Créer un fragment combiné

Pour créer un fragment combiné :

1. Dans la barre d'insertion d'objets du diagramme de séquence, cliquez sur le bouton **Fragment combiné**.
Vous pouvez associer au fragment combiné différents types d'opérateur d'interaction. La flèche située à droite du bouton offre un raccourci vers

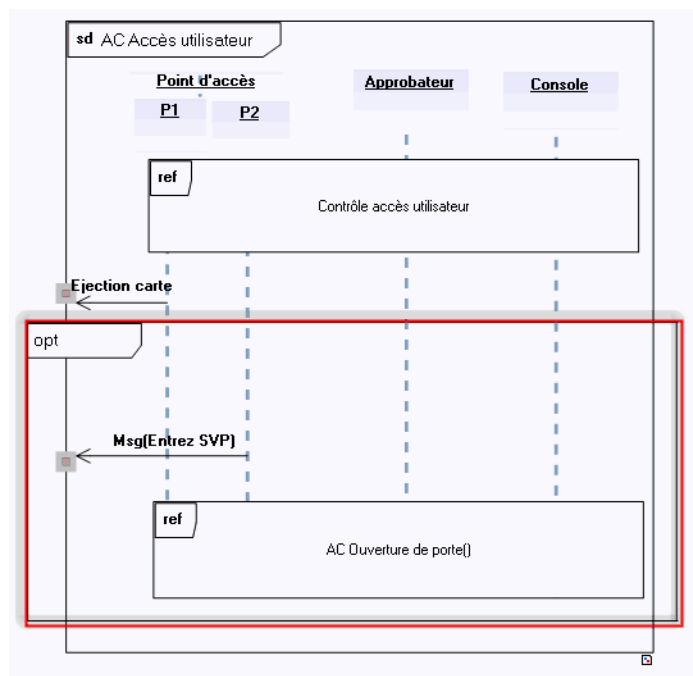
quatre d'entre eux. Voir ["Type d'opérateur d'interaction"](#), page 244.



2. Cliquez dans le diagramme.
La fenêtre de création du fragment combiné apparaît.
3. Indiquez son **Nom** et le **Type d'opérateur d'interaction** si ce n'est pas déjà fait.
4. Cliquez sur **Terminer**.

Un fragment combiné est représenté par un rectangle dont l'angle supérieur gauche affiche le type d'opérateur d'interaction.

Dans l'exemple ci-dessous, un fragment combiné de type optionnel traduit un comportement susceptible de contrarier le séquençage normal (l'ouverture de la porte).



Type d'opérateur d'interaction

Le type d'opérateur d'interaction conditionne la signification du fragment combiné. Il existe différents types d'opérateurs : seq, alt, opt, break, par, strict, loop, region, neg, assert, ignore et consider.

Alternatives

Alt exprime la possibilité de choisir entre différents comportements possibles en évaluant les conditions de garde associées à chacun des opérandes. Au plus un des opérandes pourra être exécuté.

L'opérande gardé par Else est choisi lorsqu'aucune des autres conditions n'est réalisée.

Option

Opt représente un choix entre l'unique opérande proposé ou aucun.

Arrêt (Break)

Break représente un scénario d'arrêt qui est exécuté à la place du reste du fragment d'interaction englobant.

Parallèle

Par implique que les différents opérandes peuvent être exécutés en parallèle. Les occurrences des événements des divers opérandes d'interaction peuvent être entrelacées de toutes les façons tant que l'ordre imposé par chaque opérande est préservé.

Séquence faible (Weak Sequencing)

Seq désigne un entrelacement faible entre les comportements des opérandes défini par trois propriétés :

- L'ordre des occurrences d'événements à l'intérieur de chacun des opérandes est maintenu dans le résultat.
- Les occurrences d'événements de différentes lignes de vie venant de différents opérandes peuvent apparaître dans n'importe quel ordre.
- Les occurrences d'événements d'une même ligne de vie venant de différents opérandes sont ordonnés de telle sorte que l'occurrence d'événement du premier opérande apparaît avant celle du deuxième.

Séquence stricte (Strict Sequencing)

Strict définit un séquençement strict entre les comportements des opérandes.

Négation (Negative)

Neg représente un opérande invalide.

Région critique

Critical représente une région qui doit être traitée de manière atomique, ce qui signifie que des occurrences d'événement ne peuvent pas être entrelacées avec celles de la région critique.

Ignorer / Considérer

Consider et Ignore nécessitent qu'une liste de messages pertinents soient spécifiés.

Ignorer indique que les types de certains messages sont ignorés dans le fragment combiné.

Consider signifie que seuls certains messages vont être considérés à l'intérieur du fragment combiné. C'est équivalent à définir tous les autres messages comme 'ignorés'.

Assertion

Assert représente une séquence qui est la seule valide pour un message donné.

Ainsi, toute séquence définie par un fragment d'interaction qui commence par les messages qui aboutissent à la séquence définie par le bloc assert et qui continue par un échange de messages ne respectant pas le bloc assert doit être définie comme invalide.

Les assertions sont fréquemment utilisées en combinaison avec les types Ignore et Consider.

Boucle

Loop permet d'indiquer que l'opérande d'interaction sera répété un certain nombre de fois. Il est possible de spécifier un nombre minimum et un nombre maximum de boucles, ainsi qu'une expression de continuation de la boucle.

Opérande d'interaction

Un opérande d'interaction est contenu dans un fragment combiné et représente un opérande de l'expression donnée par le fragment combiné englobant. Il peut être conditionné par une contrainte d'interaction qui sert de condition de garde.

Créer un opérande d'interaction

Pour créer un opérande d'interaction :

1. Faites un clic droit sur le fragment combiné qui contient l'opérande d'interaction.
2. Sélectionnez **Nouveau > Opérande d'interaction**.
3. Nommez l'opérande et cliquez sur **OK**.

Créer une contrainte d'interaction

Pour créer la contrainte d'interaction qui va conditionner l'opérande :

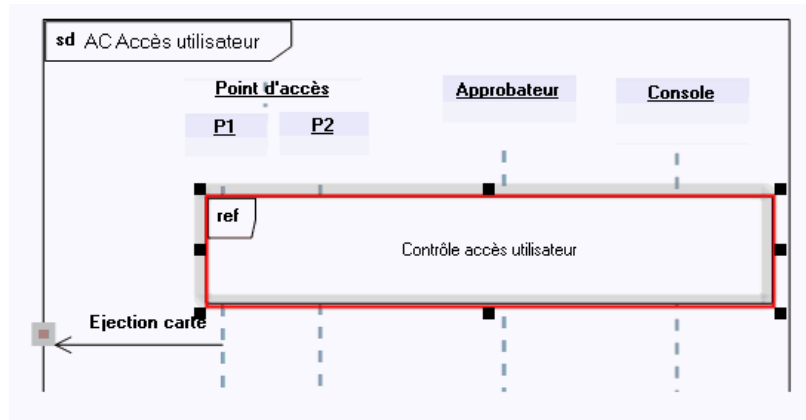
1. Ouvrez la fenêtre de propriétés de l'opérande d'interaction.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Dans le cadre **Condition**, cliquez sur **Nouveau**.

4. La condition est représentée par une contrainte. Définissez la contrainte et cliquez sur **OK**.


Utilisation d'interaction

Une utilisation d'interaction se réfère à une interaction. C'est un moyen de copier le contenu de l'interaction référencée à l'endroit de l'occurrence d'interaction.

Exemple



Pour créer une utilisation d'interaction :

1. Cliquez sur le bouton **Utilisation d'interaction** .
2. Cliquez dans le diagramme.
3. Dans la fenêtre qui apparaît, indiquez son nom et l'interaction appelée.
4. Cliquez sur **Terminer**.

Vous pouvez préciser les arguments d'une utilisation d'interaction. Un argument est une valeur spécifique correspondant à un paramètre de l'interaction appelée. Aussi, une fois l'argument créé sur l'utilisation d'interaction, vous devez le mettre en correspondance avec le paramètre de l'interaction appelée.

Pour créer un argument :

1. Ouvrez la fenêtre de propriétés de l'utilisation d'interaction.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Dans le cadre **Arguments**, cliquez sur le bouton **Nouveau**.
Une spécification de valeur est créée.

Vous pouvez la renommer et préciser ses caractéristiques en ouvrant sa fenêtre de propriétés.

Pour mettre l'argument en correspondance avec le paramètre de l'interaction appelée :

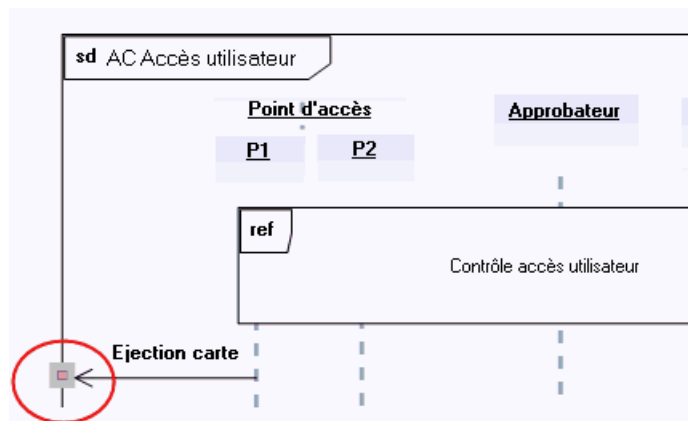
1. Dans la fenêtre de propriétés de l'utilisation d'interaction, cliquez sur l'onglet **Caractéristiques**.

2. Cliquez sur la flèche située à l'extrémité du champ **Interaction called** et sélectionnez **Modifier**.
Une fenêtre affiche les paramètres de l'interaction appelée.
3. Pour chaque paramètre, cliquez dans la colonne valeur et sélectionnez la spécification de valeur qui lui correspond.


Porte

Une porte (gate) est un point de connexion entre un message extérieur à un fragment d'interaction et un message appartenant à ce fragment d'interaction.

Exemple



Pour créer une porte dans le diagramme de séquence :

1. Cliquez sur le bouton **Porte**  de la barre d'insertion d'objets.
2. Cliquez sur le cadre délimitant l'interaction, là où vous souhaitez positionner la porte.
La porte apparaît dans le diagramme.

Continuation

Une continuation est un moyen syntaxique de définir le prolongement des séquences de différentes branches de fragment combiné alternatif. Les continuations sont similaires à des labels représentant des points intermédiaires dans un flot de contrôle.

LE DIAGRAMME DE COMMUNICATION

Le diagramme de communication est une représentation simplifiée d'un diagramme de séquence ; il se concentre sur les échanges de messages entre les objets au sein d'une interaction.

Le diagramme de séquence et le diagramme de communication sont isomorphes. Lorsqu'un diagramme de communication porte sur une interaction déjà décrite dans un diagramme de séquence, il est automatiquement initialisé à partir des informations contenues dans le diagramme de séquence.

Exemple

Diagramme de séquence

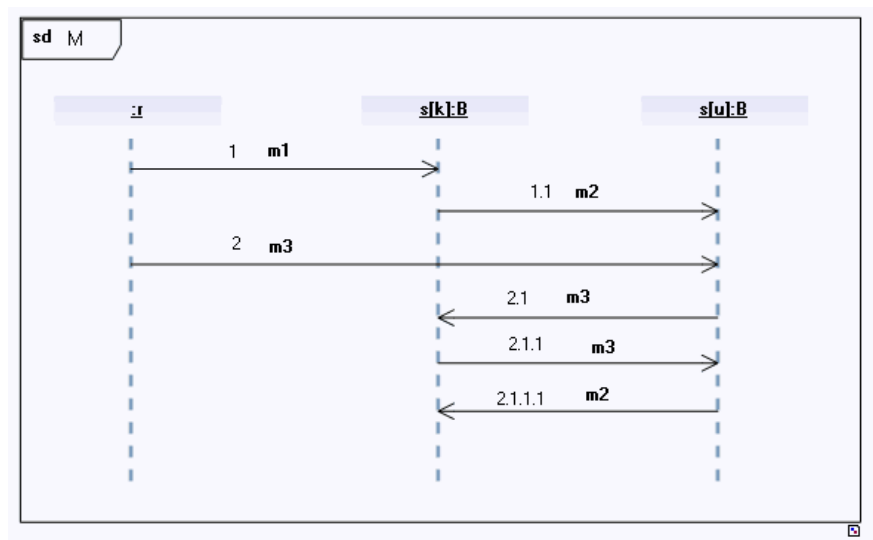
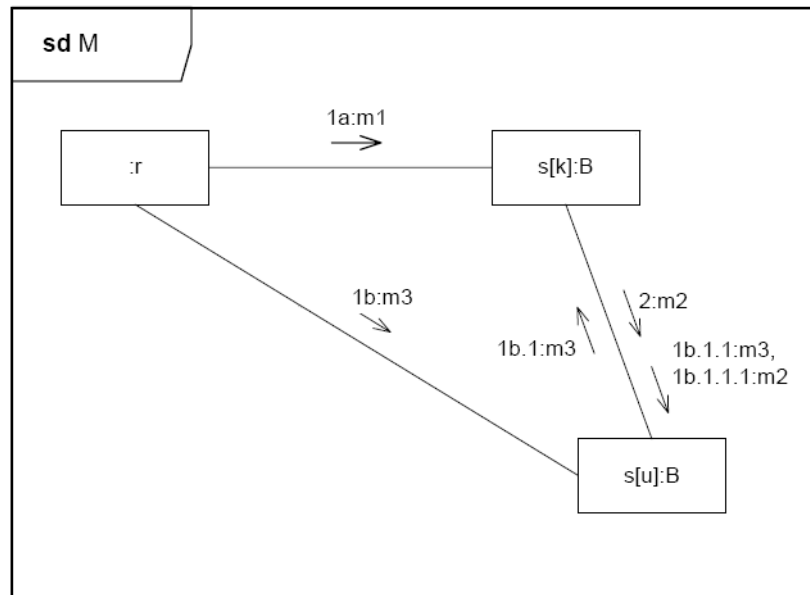



Diagramme de communication



Objets du diagramme

Les objets du diagramme de communication sont les lignes de vie et les messages transmis par des connecteurs.

Lorsque vous reliez deux lignes de vie par un connecteur , la boîte de création du connecteur propose les messages susceptibles d'être transmis.

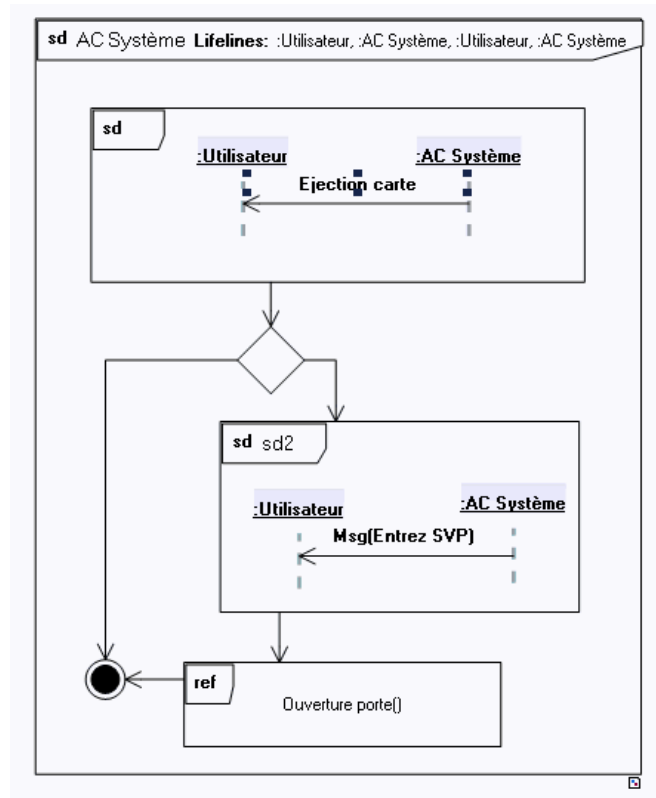
Une fois le connecteur créé, vous pouvez également lui associer de nouveaux messages via sa fenêtre de propriétés, sous l'onglet **Message**.

La séquence des messages est donnée par un numéro de séquence associé aux messages. Voir "[Calcul des numéros de séquence](#)", page 241.

Pour plus de détails sur les connecteurs, voir "[Les connecteurs](#)", page 210.

LE DIAGRAMME DE VUE GÉNÉRALE D'INTERACTION

Le diagramme de vue générale d'interaction permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences. Il vise à fournir une vue d'ensemble du flux de contrôle.



Les objets représentés dans le diagramme de vue générale d'interaction sont les interactions et utilisations d'interaction, les lignes de vie, les messages, les noeuds de contrôle et les flux de contrôle.

LE DIAGRAMME DE DÉPLOIEMENT



Le diagramme de déploiement complète le diagramme de composants en mettant en avant les ressources matérielles sur lesquelles s'exécutent les composants.

✓ ["Présentation du diagramme de déploiement", page 252.](#)

PRÉSENTATION DU DIAGRAMME DE DÉPLOIEMENT

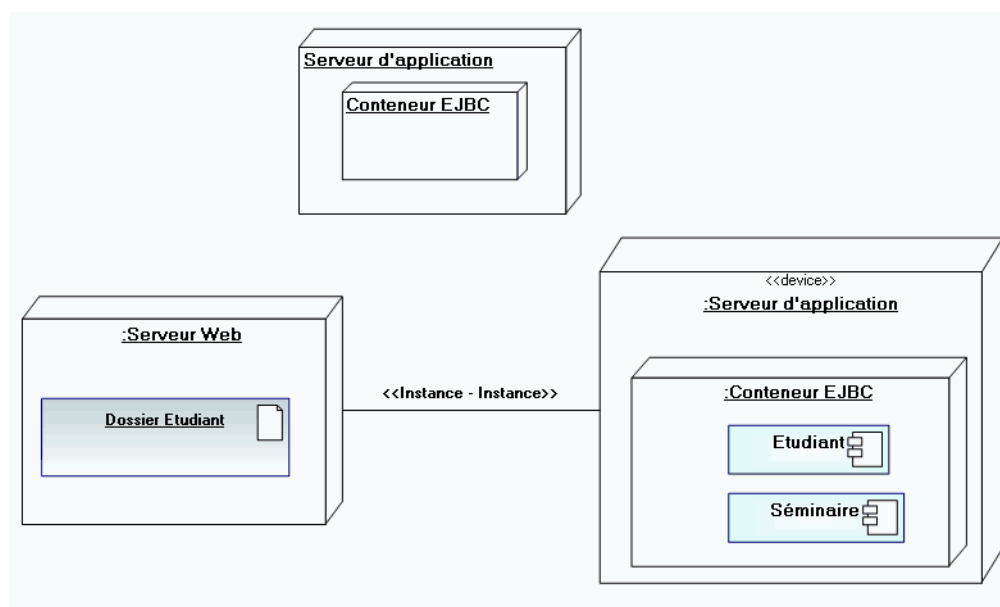
Le diagramme de déploiement complète le diagramme de composants. Il décrit les ressources matérielles (ordinateur, routeur etc.) qui composent le système et montre la répartition des composants sur ces matériels.

Il décrit également les connexions entre les composants ou les nœuds.

Ce diagramme permet également de préciser les interfaces requises et implémentées pour l'enchaînement des composants.

Il peut être illustré et complété par l'ajout d'instances de nœud, de composant ou de classe.

Exemple de diagramme de déploiement



Créer un diagramme de déploiement

Dans **HOPEX Application Design**, un diagramme de déploiement se crée à partir d'un paquetage.

Pour créer un diagramme de déploiement :

1. Faites un clic droit sur le paquetage concerné.
2. Dans le menu contextuel qui s'affiche, cliquez sur **Nouveau > Diagramme de déploiement**. je ne le vois pas

Le nouveau diagramme de déploiement s'ouvre dans la fenêtre d'édition.

Objets du diagramme de déploiement


Noeud

Un noeud est un objet physique représentant une ressource informatique disposant généralement d'une mémoire et souvent de capacités de calcul et sur lesquels des composants peuvent être déployés

Les noeuds peuvent se composer d'autre noeuds ou d'artefacts. Pour montrer qu'un composant est affecté à un nœud, il faut soit placer le composant dans le nœud, soit relier le composant au noeud par une relation de dépendance.

Voir "[Les liens de dépendance](#)", page 214.

Vous pouvez créer un noeud dans le diagramme de déploiement à l'aide du bouton

Noeud (UML)  de la barre d'insertion d'objets.

Chemin de communication

Les connexions entre noeuds sont représentées par des chemins de communication par lesquels sont échangés des signaux et des messages.


Composant

Un composant représente une partie modulaire d'un système qui encapsule son contenu et qui est remplaçable dans son environnement. Un composant définit son comportement par les interfaces qu'il fournit et celles qu'il requiert.


Un composant peut être remplacé par un autre si leurs interfaces sont conformes.

Un composant peut être un logiciel, un programme, un élément de code, etc.

Artefact

Un artefact  représente un élément d'information physique qui est utilisé ou produit par le processus de développement d'un logiciel, ou par le déploiement ou la mise en oeuvre d'un système. Ex: fichiers sources, scripts, fichiers binaires exécutables, les livrables issus d'un développement, un document produit par un traitement de texte, un message électronique, etc.

Manifestation


Une manifestation  est la restitution physique concrète dans un artefact d'un ou de plusieurs éléments de modélisation tels que des composants ou des classes.

Une dépendance de manifestation a pour source un artefact et pour cible un composant ou une classe.

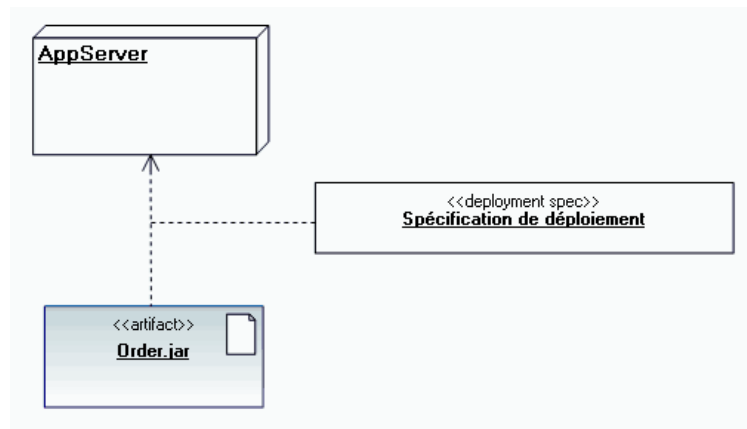
Spécification de déploiement

La spécification d'un déploiement permet d'indiquer l'ensemble des caractéristiques qui déterminent les paramètres d'exécution d'un artefact ou d'un composant déployé sur un noeud.

Configuration

Le bouton Configuration  permet de créer le lien entre une spécification de déploiement et un déploiement.

Exemple



ANNEXE : TYPE DES ATTRIBUTS



Les points suivants sont abordés ici :

- ✓ ["Types élémentaires", page 256](#)
- ✓ ["Paquetages et types élémentaires", page 258](#)
- ✓ ["Définir de nouveaux types élémentaires", page 261](#)

TYPES ÉLÉMENTAIRES

Un type élémentaire permet de mettre en commun des caractéristiques communes à plusieurs attributs. Les types élémentaires sont implémentés sous forme de classe.

Pour disposer des types élémentaires dans **HOPEX Application Design**, l'administrateur doit importer le solution pack "Information Architecture". Voir ["Conditions préalables : importer les types élémentaires", page 21.](#)

Définir un type élémentaire

Les types élémentaires sont définis dans un diagramme de classes.

Ce sont des classes pour lesquelles on précise les points suivants :

- Ce sont des classes de stéréotype "Type élémentaire".
- Ce sont des classes "Abstraites" car elles ne sont pas destinées à être instanciées.
- Ce sont des classes "Non persistantes". Elles ne doivent en effet pas donner lieu à une table dans la base de données.

Pour indiquer le type des attributs des classes :

1. Dans la fenêtre de propriétés de la classe, cliquez sur la page **Caractéristiques internes**.
2. Dépliez la section **Attributs**.
3. Cliquez dans le champ **Type expression** et sélectionnez le type de l'attribut à l'aide de la flèche.

Les classes proposées en standard sont :

Types alphanumériques		Compléments
M-Char	Chaîne de caractères alphanumériques de taille fixe	Longueur
M-Varchar	Chaîne de caractères alphanumériques de taille variable	
Types numériques		
M-Numeric	Numérique	Longueur, Décimale
M-Amount	Montant exprimé en monnaie	Longueur, Décimale
Types dates		
M-Date	Date	

M-Time	Heure	
M-Datetime	Date et heure	
Types binaires		
M-Timestamp	Identification générée automatiquement à partir de la date et de l'heure exprimée en millièmes de secondes après le 01 Janvier 1970	
M-Bool	Booléen valant 0 ou 1	
M-Multimedia	Chaîne binaire	

PAQUETAGES ET TYPES ÉLÉMENTAIRES

Paquetages



Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.

L'affectation des classes à des paquetages est extrêmement structurante. En effet, comme une classe ne peut être détenue que par un seul paquetage, il est nécessaire de définir des liens clients - fournisseurs entre les paquetages de façon à ce que les paquetages qui en ont besoin puissent utiliser les classes dont ils ne sont pas détenteurs.

Ceci est particulièrement important pour les classes de type élémentaire puisqu'elles vont être utilisées pour définir les attributs des autres classes.



Règle : Une classe est détenue par un paquetage et un seul.

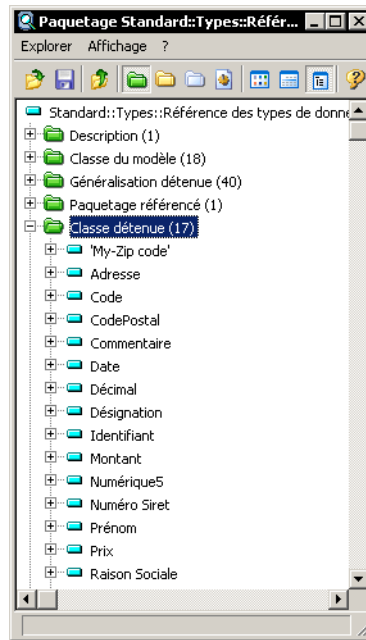
Les types élémentaires disponibles pour les attributs d'une classe dépendent du paquetage qui la détient.

Pour typer les attributs d'une classe, on proposera les types élémentaires définis pour le paquetage qui contient cette classe.

Les types élémentaires disponibles sont les classes publiques de stéréotype "Type élémentaire" détenues ou utilisées par ce paquetage ou par les paquetages dont il est client.

On peut ainsi définir un paquetage de référence (ou plusieurs) détenant les types élémentaires utilisés dans l'entreprise. Chacun des autres paquetages sera déclaré client du paquetage de référence des types élémentaires.

Dans l'exemple ci-dessous, le paquetage "Référence des types de données" détient les classes "Adresse", "Code", "Date", etc.



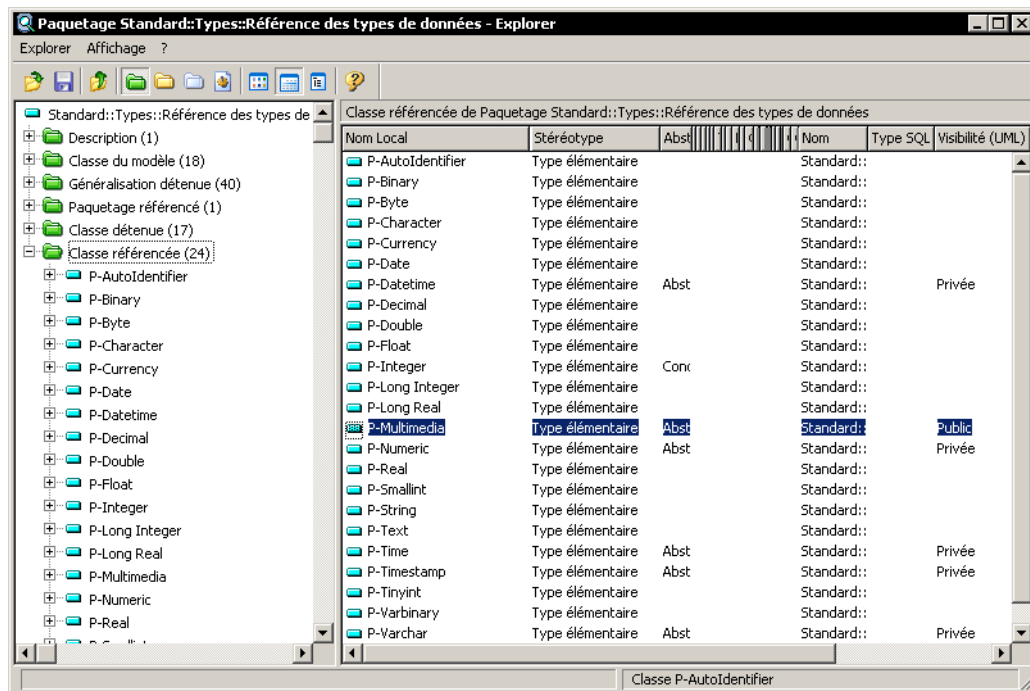
Il est référencé par les paquetages "Bibliothèque", "Gestion des commandes" etc.

Les attributs des classes de ces paquetages peuvent donc être typés à l'aide des types "Adresse", "Code", "Date", etc.

Il est également possible de préciser directement qu'un paquetage référence une classe détenue par un autre paquetage.

Dans l'exemple ci-dessous, les classes "P-Datetime", "P-Multimedia", "P-Numeric", etc., sont référencées par le paquetage "Référence des types de données" sans qu'il les détienne.

Parmi celles-ci, seule la classe "M-Multimedia" est rendue publique par ce paquetage.



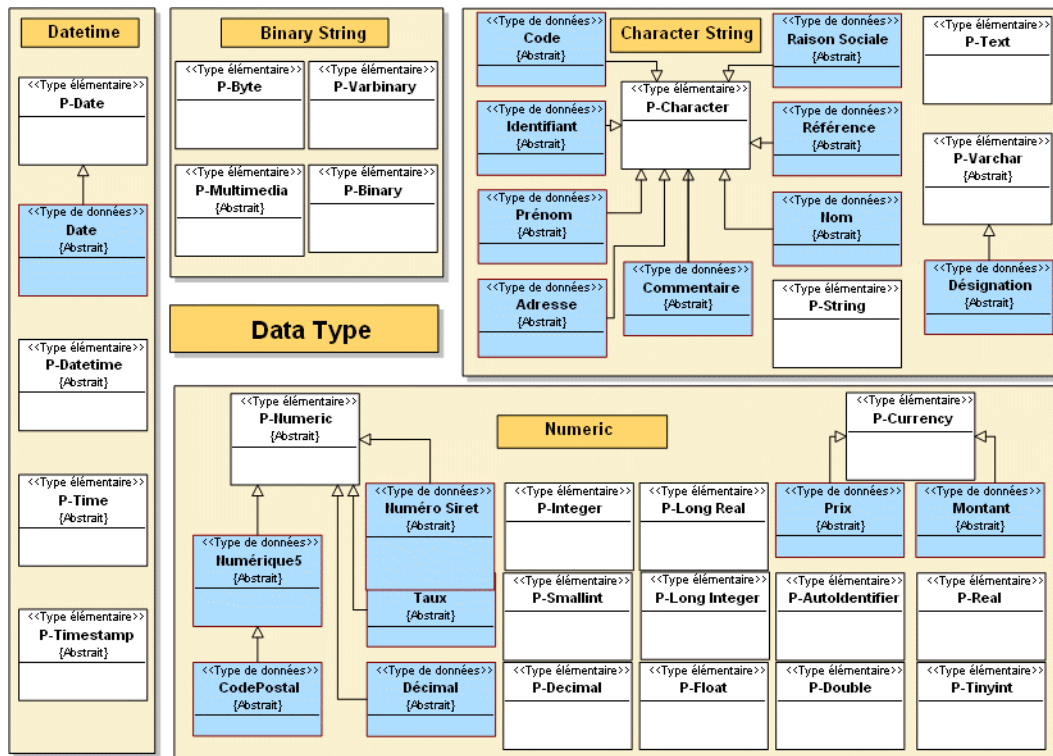
DÉFINIR DE NOUVEAUX TYPES ÉLÉMENTAIRES

De nouveaux types élémentaires peuvent être définis à l'aide d'un diagramme de classes.

Ce diagramme de classes pourra décrire, selon que l'on aura choisi ou non la structuration des classes dans des paquetages :

- Une base de données de référence.
- Le packaging des types de référence.

Vous pouvez définir vos propres types élémentaires en les déclarant sous-classes des types élémentaires proposés en standard comme dans l'exemple ci-dessous :

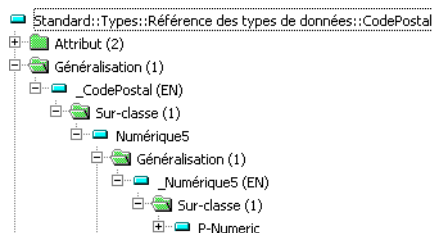


Les types élémentaires définis comme sous-classes vont hériter automatiquement des caractéristiques de leur super-classe. En particulier, la règle de transformation en datatype de la super-classe est appliquée à la sous-classe.

Il est possible de préciser sur la sous-classe une longueur et un nombre de décimales. Ceux-ci seront pris en compte pour la génération des datatypes s'ils n'ont pas déjà été définis pour la super-classe.

L'héritage peut se faire sur plusieurs niveaux.

Dans l'exemple suivant, le type élémentaire "CodePostal" est une spécialisation du type "Numérique5" de longueur 5, lui-même spécialisation du type standard "P-Numeric".

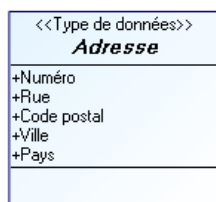


Si le nouveau type élémentaire n'est pas défini directement ou indirectement comme sous-classe d'un type élémentaire standard, il est nécessaire de mettre à jour le tableau de conversion des types élémentaires en datatypes de colonnes.

☛ Une correspondance peut également être définie directement entre un type et le datatype SQL généré pour chaque SGBD cible sans utiliser le mécanisme d'héritage (voir "Correspondances entre types pivots et datatypes" dans le guide **HOPEX Database Builder**).

Type élémentaire composé

On peut définir un type élémentaire composé en lui précisant une liste d'attributs.



Ici le type Adresse est composé du numéro, de la rue, du code postal, de la ville et du pays.

Un attribut de type Adresse donnera lieu lors de la dérivation à ces cinq colonnes.

Il est possible de décomposer un type à plusieurs niveaux en affectant un type décomposé à l'un de ses attributs.

Par exemple, on peut décomposer le code postal en code ville et code département :

