

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : <b>17 novembre 2003.</b>
	<b>Diffusion restreinte</b>	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

<b>Suivi des versions-révisions et des validations du document.</b>			
<p>Ce document annule et remplace tout document diffusé de version-révision antérieure.</p> <p>Dès réception de ce document, les destinataires ont pour obligation de détruire les versions-révisions antérieures, toutes les copies, et de les remplacer par cette version.</p> <p>Si les versions-révisions antérieures sont conservées pour mémoire, les destinataires doivent s'assurer qu'elles ne peuvent être confondues avec cette présente version-révision dans leur usage courant.</p>			
Version.	Date.	Auteurs.	Création, modification ou validation.
A	22 oct. 2003.	JPD.	Création.

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

# 1 Tables

## 1.1 Table des matières

<b>1</b>	<b>Tables</b>	<b>2</b>
1.1	Table des matières	2
1.2	Table des illustrations	2
<b>2</b>	<b>Références</b>	<b>3</b>
2.1	Glossaire	3
2.2	Ressources	3
<b>3</b>	<b>Introduction</b>	<b>4</b>
3.1	Objet du document	4
3.2	Audience	4
3.3	Pré-requis	5
<b>4</b>	<b>Description</b>	<b>6</b>
4.1	Objet du composant	6
4.2	Conventions	6
4.3	Principes	6
<b>5</b>	<b>Modèle de données</b>	<b>7</b>
<b>6</b>	<b>Interaction avec l'utilisateur</b>	<b>10</b>
6.1	Interfaces homme-machine	10
6.2	Editions	13
<b>7</b>	<b>Interaction du système avec les autres systèmes</b>	<b>16</b>
7.1	Appels de service	16
7.2	Echanges de données	17
<b>8</b>	<b>Traitements</b>	<b>20</b>
8.1	Enchaînement des traitements	20
8.2	Traitements complexes	21

## 1.2 Table des illustrations

Texte 1	Exemple d'objet d'un composant logiciel	6
Texte 2	Exemple de conventions d'un composant logiciel	6
Texte 3	Exemple de principes d'un composant logiciel	6
Tableau 4	Symboles du modèle physique de données	8
Diagramme 5	Exemple de modèle physique de données	8
Tableau 6	Exemple de description d'une entité	9
Tableau 7	Exemple de paramètres d'une interface homme-machine	10
Copie d'écran 8	Exemple d'une interface homme-machine d'un composant logiciel	11
Tableau 9	Exemple de description des champs d'une interface homme-machine	12
Tableau 10	Exemple de règles de contrôle de validité d'une interface homme-machine	13
Tableau 11	Exemple de paramètres d'une édition	13
Copie d'écran 12	Exemple d'une édition d'un composant logiciel	14
Tableau 13	Exemple de formatage des champs d'une édition	14
Tableau 14	Exemple de règle de contrôle de validité d'une édition	15
Tableau 15	Exemple de prototype d'un appel de service d'un composant logiciel	17
Tableau 16	Exemple de paramètres d'un échange de données	18
Tableau 17	Exemple d'échange de données d'un composant logiciel	19
Tableau 18	Exemple de règle de contrôle de validité sur un échange de données	19
Tableau 19	Symboles de l'enchaînement des traitements	21
Diagramme 20	Exemple d'algorithme d'un traitement complexe	22
Texte 21	Exemple de description des étapes critiques d'un algorithme	22

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence</b> : UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

## 2 Références

### 2.1 Glossaire

Liste des définitions des termes employés.	
Ce tableau recense tous les termes, les concepts particuliers ainsi que les abréviations employés dans ce document.	
Terme, concept, abrégé.	Définition du terme, du concept ou de l'abréviation.

### 2.2 Ressources

Liste des documents applicables et en référence.		
<p>Un document est <b>applicable</b> à partir du moment où son contenu est validé et que l'activité ou le projet fait partie de son périmètre d'application. Il est obligatoire d'appliquer son contenu.</p> <p>Un document est en <b>référence</b> à partir du moment où son contenu n'est pas validé ou que l'activité ou le projet ne fait partie de son périmètre d'application. Il est recommandé d'appliquer son contenu mais cela n'est pas obligatoire.</p> <p>Un document applicable est indiqué par <b>A1, A2, A3</b>, etc. Un document en référence est indiqué par <b>R1, R2, R3</b>, etc.</p>		
Index.	Nom du document.	Commentaire.
A1	UpComp-Plan Qualité-000005	Méthode documentaire.
A2	UpComp-Plan Qualité-000006	Processus de management de projet.
A3	UpComp-Plan Qualité-000041	Modèle de spécification technique d'un composant logiciel.
R4	UpComp-Meilleure pratique-000016	Spécification technique d'un composant logiciel.

	<b>Méthode de spécification technique d'un composant logiciel</b>	<b>Date rédaction :</b> 17 novembre 2003.
	<b>Diffusion restreinte</b>	<b>Date validation :</b>
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

## 3 Introduction

### 3.1 Objet du document

L'objet de ce document est de définir la méthode de spécification d'un composant logiciel développé en spécifique pour le projet.

Le respect de ces règles fondamentales énoncées ci-après contribue à la réussite du projet où chaque intervenant trouvera satisfaction au regard de sa contribution.

Cette spécification se base sur la réponse aux questions suivantes :

- **Description du composant.**  
Quel est son objet ? Quelles sont ses conventions ? Quels sont ses principes ?
- **Modèle de données.**  
Quel est le modèle physique des données ?
- **Interaction avec l'utilisateur.**  
Quels sont les interfaces homme-machine ou les éditions non détaillées dans la spécification des besoins du système ? Quels sont les contrôles de validité ?
- **Interaction avec les autres systèmes.**  
Quels sont les appels de service ou les échanges de données non détaillés dans la spécification des besoins du système ? Quels sont les contrôles de validité ?
- **Traitements complexes.**  
Quel est l'enchaînement des traitements ? Quels sont les algorithmes des traitements complexes ?

Ce document présente comment répondre à ces questions à l'aide d'exemples. Le **Modèle de spécification technique d'un composant logiciel** [A3] comporte trois propriétés :

- **NomDuProjet** pour le nom du projet.
- **NomDuModule** pour le nom du module.
- **NomDuComposant** pour le nom du composant.

Un exemple de l'application de la méthode est la meilleure pratique **Spécification technique d'un composant logiciel** [R4].

a A l'inverse des autres spécifications, celle-ci est écrite au fur et à mesure que le composant technique est écrit.

### 3.2 Audience

Ce document s'adresse aux :

- **Directeurs de projets et chefs de projets.**  
Pour la compréhension du composant technique.
- **Ingénieurs de développement.**  
Pour savoir comment est conçu le composant technique.

Pour aider ces personnes à remplir le document **Spécification technique d'un composant logiciel**, leur manager et la cellule de support projet se tiennent à leur disposition.

	<b>Méthode de spécification technique d'un composant logiciel</b>	<b>Date rédaction :</b> 17 novembre 2003.
	Diffusion restreinte	<b>Date validation :</b>
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

### 3.3 Pré-requis

Le pré-requis est la connaissance de la **Méthode documentaire** [A1] et le **Processus de management de projet** [A2].

	<b>Méthode de spécification technique d'un composant logiciel</b>	<b>Date rédaction :</b> 17 novembre 2003.
	Diffusion restreinte	<b>Date validation :</b>
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

## 4 Description

### 4.1 Objet du composant

Cette section décrit sommairement l'objet du composant logiciel.

Voici un exemple :

L'objet du composant logiciel **Contacts** du module **Contacts** est de présenter les contacts mémorisés dans la base de données, de rechercher un contact particulier ou d'imprimer une fiche d'un contact.

**Texte 1 – Exemple d'objet d'un composant logiciel**

### 4.2 Conventions

Cette section décrit les conventions propres au composant logiciel. Il s'agit généralement de conventions de dénomination.

Voici un exemple :

- **Le préfixe du module est *Contacts*.**  
Il est employé pour les dénominations de toutes les définitions non statiques
- **Taille des chaînes de caractère.**  
Les constantes pour les chaînes de caractère sont les suivantes :
  - **CO\_TailleIdf** pour un identificateur.
  - **CO\_TailleNomFichier** pour un nom de fichier ou de répertoire.
  - **CO\_TailleBuffer** pour une variable temporaire de construction d'une chaîne de caractère.

**Texte 2 – Exemple de conventions d'un composant logiciel**

Si le logiciel ne comporte aucune convention, la mention « Néant » est inscrite.

### 4.3 Principes

Cette section décrit les principes des algorithmes propres au composant. Il s'agit des connaissances propres à l'état de l'art, les parties spécifiques des algorithmes étant décrites dans la section intitulées **Traitements**.

Voici un exemple :

- Les interfaces hommes-machines sont réalisées avec l'**API Win32**.
- Les éditions sont réalisées avec l'**API Win32**.
- L'accès à la base de données est réalisé via l'**API Oracle Call Interface (OCI)**.

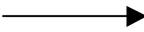
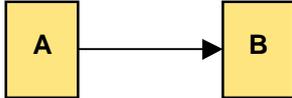
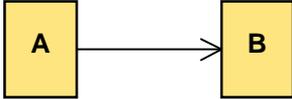
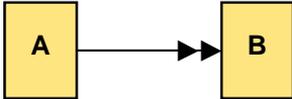
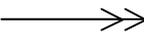
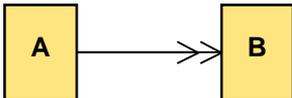
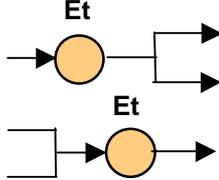
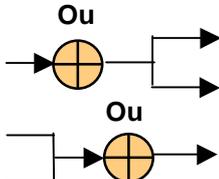
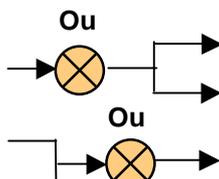
**Texte 3 – Exemple de principes d'un composant logiciel**

Si le logiciel ne repose sur aucun principe, la mention « Néant » est inscrite.

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

## 5 Modèle de données

Cette section présente le modèle physique de données propre au composant du module. Les symboles suivants sont utilisés à cette fin :

Symbole	Sémantique	Exemple
<b>Entité</b>  	<b>Entité.</b> Une classe, une structure ou une union. Dans l'exemple, il s'agit d'une adresse.	<b>Adresse</b>  
	<b>Dépendance (1,1).</b> Dans l'exemple, l'entité <b>A</b> est en relation avec une unique occurrence de l'entité <b>B</b> .	
	<b>Dépendance (0,1).</b> Dans l'exemple, l'entité <b>A</b> est en relation avec éventuellement une occurrence de l'entité <b>B</b> .	
	<b>Dépendance (1,n).</b> Dans l'exemple, l'entité <b>A</b> est en relation avec au moins une occurrence de l'entité <b>B</b> .	
	<b>Dépendance (0,n).</b> Dans l'exemple, l'interface <b>A</b> est en relation avec éventuellement une ou plusieurs occurrences de l'entité <b>B</b> .	
<b>Et</b> 	<b>Conjonction de dépendances.</b> Pour une dépendance sortante, toutes les occurrences des entités sont simultanément liées.  Pour une dépendance entrante, toutes les occurrences des entités sont simultanément liées.	
<b>Ou inclusif</b> 	<b>Disjonction inclusive de dépendances.</b> Pour une dépendance sortante, une ou plusieurs des occurrences des entités est liée.  Pour une dépendance entrante, une ou plusieurs des occurrences des entités est liée.	
<b>Ou exclusif</b> 	<b>Disjonction exclusive de dépendances.</b> Pour une dépendance sortante, une des occurrences d'une entité est liée, à l'exclusion des autres.  Pour une dépendance entrante, une des occurrences d'une entité est liée, à l'exclusion des autres.	

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

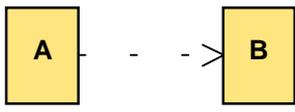
Symbole	Sémantique	Exemple
- - - >	<b>Héritage.</b> Dans l'exemple, l'entité <b>A</b> hérite des propriétés et des méthodes de l'entité <b>B</b> .	

Tableau 4 – Symboles du modèle physique de données

Voici un exemple :

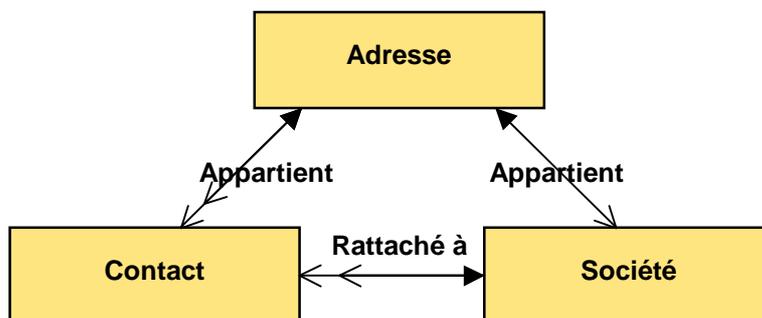


Diagramme 5 – Exemple de modèle physique de données

Si un élément du modèle est décrit ailleurs, dans une interface technique commentée par exemple, seul le diagramme du modèle est utile, agrémenté des commentaires nécessaires à sa compréhension.

Si un élément du modèle n'est pas décrit ailleurs, pour une table d'une base de données ou un fichier structuré par exemple, le diagramme du modèle est complété par la description complète de l'élément.

Voici un exemple pour l'entité **Contact** :

Propriétés.		
Dénomination.	Type.	Valeurs possibles.
<b>Description.</b>		
<b>Batiment.</b>	Varchar(255).	« Bâtiment A ».
Complément de l'adresse, tel le nom du bâtiment, le numéro de l'escalier, etc.		
<b>CodePostal.</b>	Number(7).	75001.
Numéro du code postal.		
<b>Identifiant.</b>	Number(7).	1.
Clé unique d'identification de l'adresse.		
<b>IdentifiantContact.</b>	Nul Ou Number(7).	1.
Clé unique d'identification du contact si l'adresse est celle d'un contact.		
<b>IdentifiantSociete.</b>	Nul Ou Number(7).	1.
Clé unique d'identification de la société si l'adresse est celle d'une société.		
<b>Numero.</b>	Number(7,2).	1.
Numéro de la rue.		

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

<b>Rue.</b>	Varchar(255).	« Boulevard Magenta ».
Nom de la rue, de l'avenue ou du boulevard.		
<b>Ville.</b>	Varchar(255).	« Paris ».
Nom de la ville.		
<b>Méthodes.</b>		
<b>Dénomination et paramètres.</b>		
<b>Description.</b>		
<b>Creer().</b>		
Création d'une nouvelle adresse.		
<b>Lire(Critere : Varchar(255)).</b>		
Lecture d'un ou de plusieurs adresses en fonction du critère <b>Critere</b> .		
<b>Modifier().</b>		
Modification d'une adresse existante.		
<b>Supprimer().</b>		
Suppression d'une adresse existante.		

**Tableau 6 – Exemple de description d'une entité**

Voici le contenu de chaque colonne de ce tableau :

- **Pour les propriétés.**  
Elles sont énumérées en premier, par ordre alphabétique afin d'en rechercher une facilement.
  - **La dénomination.**  
Il s'agit du nom de la propriété composé d'un ou plusieurs mots.
  - **Le type.**  
Il s'agit du nom du type de la propriété.
  - **Les valeurs possibles.**  
Nous donnons un exemple de la valeur possible de la propriété. Pour un type **Caractère**, la valeur est écrite entre caractères **guillemets « »**.
  - **La description.**  
La description permet de préciser la sémantique de la propriété.
- **Pour les méthodes.**  
Elles sont énumérées en second, par ordre alphabétique afin d'en rechercher une plus facilement.
  - **La dénomination et les paramètres.**  
Il s'agit du nom de la méthode composé d'un ou plusieurs mots dont un verbe à l'infinitif. Les paramètres obligatoires sont écrits entre parenthèses et séparés par une virgule.
  - **La description.**  
La description permet de préciser la sémantique de la méthode. Les paramètres sont alors précisés.

**M**

A l'inverse du modèle métier, les propriétés correspondant aux relations sont décrites.

	<b>Méthode de spécification technique d'un composant logiciel</b>	<b>Date rédaction :</b> 17 novembre 2003.
	Diffusion restreinte	<b>Date validation :</b>
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

## 6 Interaction avec l'utilisateur

### 6.1 Interfaces homme-machine

Cette section décrit les interfaces homme-machine propres au composant qui ne sont pas spécifiées dans le document *Spécification des besoins du système* ou les informations complémentaires de celles qui y sont décrites.

La description de chaque paramètre de l'interface homme-machine est mentionnée dans le tableau suivant :

Paramètres.	Descriptions.
IdentificationAdresse.	Adresse optionnelle du contact ou de la société à modifier.

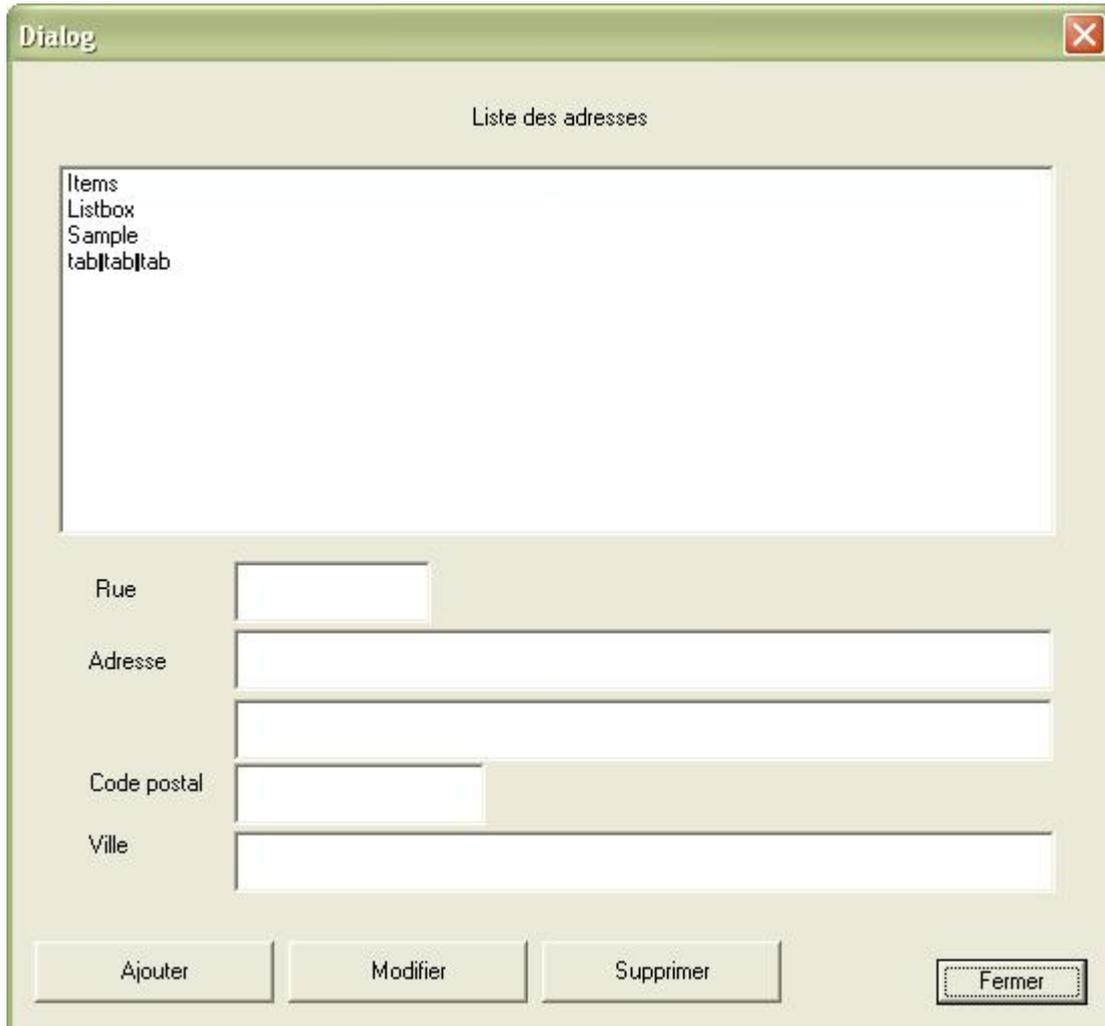
Tableau 7 – Exemple de paramètres d'une interface homme-machine

Voici le contenu de chaque colonne de ce tableau :

- **Descriptions.**  
Description du paramètre.
- **Paramètres.**  
Noms des paramètres de l'interface homme-machine.

La description de l'interface homme-machine s'appuie sur sa copie d'écran. Voici un exemple :

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		



Copie d'écran 8 – Exemple d'une interface homme-machine d'un composant logiciel

Voici un exemple pour l'interface homme-machine **Adresse** :

Champ.	Accès.	Format.	Style.
<b>Description.</b>			
<b>Expression.</b>			
<b>Batiment.</b>	Ecriture.	Normal.	Normal.
Complément du nom de la rue de l'adresse sélectionnée.			
Batiment.			
<b>CodePostal.</b>	Ecriture.	Normal.	Normal.
Numéro du code postal.			
Caractere(CodePostal, "#####").			
<b>Liste des</b>	Lecture.	Normal.	Normal.

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

<b>adresses.</b>			
Présentation d'une adresse complète.			
Caractere(Numero, "&&&#") + ", " + Rue + "-" + Batiment + "-" + Caractere(CodePostal, "#####")+ " " + Ville .			
<b>Numero.</b>	Ecriture.	Normal.	Normal.
Numéro de la rue de l'adresse sélectionnée.			
Numero.			
<b>Rue.</b>	Ecriture.	Normal.	Normal.
Nom de la rue de l'adresse sélectionnée.			
Rue.			
<b>Ville.</b>	Ecriture.	Normal.	Normal.
Nom de la ville de l'adresse sélectionnée.			
Ville.			

**Tableau 9 – Exemple de description des champs d'une interface homme-machine**

Voici la description de chaque champ de l'interface homme-machine :

- **Accès.**  
Mode d'accès au champ pour une zone de saisie :
  - **Ecriture.**  
Il est possible de saisir une première valeur, si le profil le permet.
  - **Lecture.**  
Il n'est pas possible de la modifier, indépendamment du profil.
  - **Lecture / écriture.**  
Il est possible de modifier la valeur existante, si le profil le permet.
- **Champs.**  
Nom du champ correspondant à libellé, une zone de texte, etc. Le tableau est trié sur cette colonne de la sorte à chercher facilement un champ.
- **Descriptions.**  
Description littérale du champ.
- **Expressions.**  
Expression dans le langage de programmation utilisé – **C / C++** ou **Up! 5GL** – donnant sa valeur au champ.
- **Formats.**  
Format du champ.
  - **Pour un booléen.**  
Libre uniquement.
  - **Pour un énuméré.**  
Libre uniquement.
  - **Pour un entier.**  
Filtre numérique. **&** représente un chiffre facultatif, **#** un chiffre obligatoire, etc. Pour plus de précisions, voir la fiche du type **Caractere d'Up! Kernel**.

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

- **Pour un réel.**  
Filtre numérique. **&** représente un chiffre facultatif, **#** un chiffre obligatoire, etc. Pour plus de précisions, voir la fiche du type **Caractere** d'**Up ! Kernel**.
- **Pour une date.**  
Filtre de date. **J** représente le jour, **M** représente le mois, etc. Pour plus de précisions, voir la fiche du type **Caractere** d'**Up ! Kernel**.
- **Pour un caractère.**  
Libre pour la valeur sans transformation, majuscule, minuscule.
- **Styles.**  
Style de présentation du champ ou attribut d'écriture – Gras, italique, souligné, etc.

a L'interface homme-machine peut éventuellement comporter des règles de contrôle de validité.  
Voici un exemple pour l'interface homme-machine **Adresse** :

Nom.	Description.
R1.	Le code postal est obligatoire.
R2.	Le numéro de la rue est obligatoire.
R3.	La rue est obligatoire.
R4.	La ville est obligatoire.
R5.	Le code postal doit commencer par un numéro de département.

Tableau 10 – Exemple de règles de contrôle de validité d'une interface homme-machine

Si le composant ne comporte aucune interface homme-machine spécifique ou si aucun complément n'est apporté, la mention « Néant » est inscrite.

## 6.2 Editions

Cette section décrit les éditions propres au composant qui ne sont pas spécifiées dans le document **Spécification des besoins du système** ou les informations complémentaires de celles qui y sont décrites.

La description de chaque paramètre de l'édition est mentionnée dans le tableau suivant :

Paramètres.	Descriptions.
IdentifiantContact.	Paramètre optionnel de l'identifiant du contact.

Tableau 11 – Exemple de paramètres d'une édition

Voici le contenu de chaque colonne de ce tableau :

- **Descriptions.**  
Description du paramètre.
- **Paramètres.**  
Noms des paramètres de l'édition.

La description de l'édition s'appuie sur sa copie d'écran. Voici un exemple :

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		



Copie d'écran 12 – Exemple d'une édition d'un composant logiciel

Voici un exemple de formatage des champs :

Champ.	Format.	Style.
<b>Description.</b>		
<b>Expression.</b>		
<b>Nom.</b>	Majuscule.	Normal.
Nom du contact.		
Nom.		
<b>Prénom.</b>	Minuscule.	Gras.
Prénom du contact.		
Prenom.		

Tableau 13 – Exemple de formatage des champs d'une édition

Voici le contenu de chaque colonne de ce tableau :

- **Champs.**  
Nom du champ correspondant à libellé, une zone de texte, etc. Le tableau est trié sur cette colonne de la sorte à chercher facilement un champ.
- **Descriptions.**  
Description littérale du champ.

	<b>Méthode de spécification technique d'un composant logiciel</b>	<b>Date rédaction :</b> 17 novembre 2003.
	<b>Diffusion restreinte</b>	<b>Date validation :</b>
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

- **Expressions.**  
Expression dans le langage de programmation utilité – **C / C++** ou **Up ! 5GL** – donnant sa valeur au champ.
- **Formats.**  
Format du champ.
  - **Pour un booléen.**  
Libre uniquement.
  - **Pour un énuméré.**  
Libre uniquement.
  - **Pour un entier.**  
Filtre numérique. **&** représente un chiffre facultatif, **#** un chiffre obligatoire, etc. Pour plus de précisions, voir la fiche du type **Caractere d'Up ! Kernel.**
  - **Pour un réel.**  
Filtre numérique. **&** représente un chiffre facultatif, **#** un chiffre obligatoire, etc. Pour plus de précisions, voir la fiche du type **Caractere d'Up ! Kernel.**
  - **Pour une date.**  
Filtre de date. **J** représente le jour, **M** représente le mois, etc. Pour plus de précisions, voir la fiche du type **Caractere d'Up ! Kernel.**
  - **Pour un caractère.**  
Libre pour la valeur sans transformation, majuscule, minuscule.
- **Styles.**  
Style de présentation du champ ou attribut d'écriture – Gras, italique, souligné, etc.

a L'édition peut éventuellement comporter des règles de contrôle de validité.

Voici un exemple pour l'édition **Liste des contacts** :

Nom.	Description.
R1.	Si le paramètre <b>IdentifiantContact</b> n'est pas renseigné, le listing de tous les contacts est édité.
R2.	Si le paramètre <b>IdentifiantContact</b> est renseigné, la fiche du contact demandé est éditée.

Tableau 14 – Exemple de règle de contrôle de validité d'une édition

Si le composant ne comporte aucune édition spécifique ou si aucun complément n'est apporté, la mention « Néant » est inscrite.

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

## 7 Interaction du système avec les autres systèmes

### 7.1 Appels de service

Cette section décrit les appels de service propres au composant qui ne sont pas spécifiés dans le document **Spécification des besoins du système** ou les informations complémentaires de ceux qui y sont décrits.

Pour chaque appel de service, il s'agit de définir :

- **La description du service.**  
Elle précise la fonction remplie par le service.
- **La volumétrie estimée.**  
En nombre moyen d'appels sur une période assorti de son écart-type.
- **La qualité de service attendue.**  
Quels sont le calendrier et les créneaux horaires de disponibilité du service ?

Voici la description du prototype d'appel du service **Consultation des adresses** :

Champ.	Type.	Mode de passage.
<b>Description.</b>		
<b>Expression.</b>		
<b>Batiment.</b>	<b>Nul Ou</b> Varchar(255).	Sortie.
Complément au nom de la rue.		
Batiment.		
<b>CodePostal.</b>	<b>Nul Ou</b> Number(5).	Sortie.
Code postal de l'adresse demandée.		
CodePostal..		
<b>IdentifiantContact.</b>	Number(7).	Entrée.
Identifiant du contact pour lequel l'adresse est demandée.		
Identifiant.		
<b>Numero.</b>	<b>Nul Ou</b> Number(7).	Sortie.
Numéro de la rue de l'adresse demandée.		
Numero.		
<b>Rue.</b>	<b>Nul Ou</b> Varchar(255).	Sortie.
Nom de la rue.		
Rue de l'adresse demandée.		
<b>Ville.</b>	<b>Nul Ou</b> Varchar(255).	Sortie.
Nom de la ville de l'adresse demandée.		

	<b>Méthode de spécification technique d'un composant logiciel</b>	<b>Date rédaction :</b> 17 novembre 2003.
	<b>Diffusion restreinte</b>	<b>Date validation :</b>
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		
Ville.		

**Tableau 15 – Exemple de prototype d'un appel de service d'un composant logiciel**

Voici le contenu de chaque colonne de ce tableau :

- **Champs.**  
Nom du champ correspondant à valeur échangée.
- **Descriptions.**  
Description littérale du champ.
- **Expressions.**  
Expression dans le langage de programmation utilisé – **C / C++** ou **Up! 5GL** – donnant sa valeur au champ.
- **Modes de passage.**  
Mode de passage de la valeur échangée :
  - **Entrée.**  
La valeur est passée en paramètre de l'appel du service. L'appel de service ne peut pas la modifier.
  - **Entrée / sortie.**  
La valeur est passée en paramètre de l'appel de service et elle est récupérée en résultat. L'appel de service l'a peut-être modifiée.
  - **Sortie.**  
La valeur est récupérée en résultat de l'appel du service. L'appel de service doit la renseigner si elle est obligatoire.
- **Types.**  
Type du champ exprimé de façon identique au type d'une propriété d'une interface métier.

**a** L'appel de service peut éventuellement comporter des règles de contrôle de validité.

Si le composant ne comporte aucun appel de service spécifique ou si aucun complément n'est apporté, la mention « Néant » est inscrite.

## 7.2 Echanges de données

Cette section décrit les échanges de données propres au composant qui ne sont pas spécifiés dans le document **Spécification des besoins du système** ou les informations complémentaires de ceux qui y sont décrits.

Pour chaque échange de données, il s'agit de définir :

- **La description de l'échange.**  
Elle précise l'objet de l'échange de données.
- **La volumétrie estimée.**  
En nombre moyen d'enregistrements par échange assorti de son écart-type.
- **La fréquence attendue.**  
Sur une période donnée.
- **Le temps de traitement attendu.**  
Pour une volumétrie moyenne.

La description de chaque paramètre de l'échange de données est mentionnée dans le tableau suivant :

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

Paramètres.	Descriptions.
<b>Fichier.</b>	Nom du fichier contenant les enregistrements à charger.
<b>Rejet.</b>	Nom du fichier contenant les enregistrements rejetés.

**Tableau 16 – Exemple de paramètres d'un échange de données**

Voici le contenu de chaque colonne de ce tableau :

- **Descriptions.**  
Description du paramètre.
- **Paramètres.**  
Noms des paramètres de l'échange de données.

Voici la description du format de l'échange **Chargement des contacts** :

Champ.	Type.
<b>Description.</b>	
<b>Expression.</b>	
<b>Batiment.</b>	Varchar(255).
Complément à l'adresse du contact.	
Adresse.Batiment.	
<b>Code postal.</b>	Number(6).
Code postal de l'adresse du contact.	
Adresse.CodePostal.	
<b>e-Mail.</b>	Varchar(255).
Adresse e-mail du contact.	
EMail.	
<b>Nom.</b>	Varchar(255).
Nom du contact.	
Nom.	
<b>Numero.</b>	Number(4).
Numéro de la rue de l'adresse du contact.	
Adresse.Numero.	
<b>Prénom.</b>	Varchar(255).
Prénom du contact.	
Prenom.	
<b>Raison sociale.</b>	Varchar(255).
Raison sociale de la société du contact.	
Societe.RaisonSociale.	
<b>Rue.</b>	Varchar(255).

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence</b> : UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

Nom de la rue de l'adresse du contact.	
Adresse.Rue.	
<b>Siren.</b>	Number(9).
Numéro de <b>Siren</b> de la société du contact.	
Societe.Siren.	
<b>Téléphone.</b>	Number(8).
Numéro de téléphone du contact.	
Telephone.	
<b>Ville.</b>	Varchar(255).
Ville de l'adresse du contact.	
Ville.	

**Tableau 17 – Exemple d'échange de données d'un composant logiciel**

Voici le contenu de chaque colonne de ce tableau :

- **Champ.**  
Nom du champ correspondant à valeur échangée.
- **Description.**  
Description littérale du champ.
- **Type.**  
Type du champ exprimé de façon identique au type d'une propriété d'une interface métier.

L'échange de données peut éventuellement comporter des règles de contrôle de validité.

Voici un exemple pour l'échange de données **Changement des contacts** :

Nom.	Description.
R1.	Le code postal est obligatoire.
R2.	Le nom est obligatoire.
R3.	Le prénom est obligatoire.
R4.	Le numéro de rue est obligatoire.
R5.	Le nom de la rue est obligatoire.
R6.	La raison sociale est obligatoire.
R7.	Le <b>Siren</b> est obligatoire.
R8.	La ville est obligatoire.
R9.	Il ne doit pas y avoir de contradiction dans les valeurs de ( <b>Raison sociale, Siren</b> ) entre celles du fichier et celles dans la base de données.

**Tableau 18 – Exemple de règle de contrôle de validité sur un échange de données**

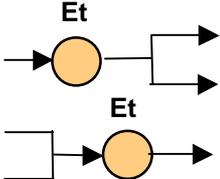
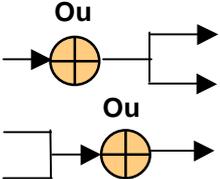
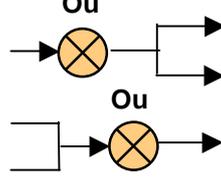
Si le composant ne comporte aucun échange de données spécifique ou si aucun complément n'est apporté, la mention « Néant » est inscrite.

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

## 8 Traitements

### 8.1 Enchaînement des traitements

Cette section présente l'enchaînement global des principaux traitements du composant et l'interaction de ceux-ci avec les traitements des autres composants. Les symboles suivants sont utilisés à cette fin :

Symbole	Sémantique	Exemple
	<b>Début ou fin de l'enchaînement.</b> Il y a début s'il y a un enchaînement sortant. Il y a fin s'il y a un enchaînement entrant.	
<div style="border: 1px solid black; background-color: yellow; padding: 5px; width: fit-content; margin: auto;"> <b>Traitement</b> </div>	<b>Traitement.</b> Un algorithme particulier du composant. Dans l'exemple, il s'agit de trier une liste.	<div style="border: 1px solid black; background-color: yellow; padding: 5px; width: fit-content; margin: auto;"> <b>TrierListe()</b> </div>
<div style="border: 1px solid black; background-color: gray; padding: 5px; width: fit-content; margin: auto;"> <b>Traitement</b> </div>	<b>Traitement.</b> Un algorithme particulier d'un autre composant. Dans l'exemple, il s'agit de trier une liste.	<div style="border: 1px solid black; background-color: gray; padding: 5px; width: fit-content; margin: auto;"> <b>TrierListe()</b> </div>
	<b>Enchaînement.</b> Enchaînement de deux algorithmes. Dans l'exemple, l'algorithme <b>A</b> s'enchaîne sur l'algorithme <b>B</b> .	<div style="display: flex; align-items: center; gap: 20px;"> <div style="border: 1px solid black; background-color: yellow; padding: 5px;">A</div> <div style="font-size: 2em;">→</div> <div style="border: 1px solid black; background-color: yellow; padding: 5px;">B</div> </div>
<b>Et</b> 	<b>Conjonction d'enchaînements.</b> Pour un enchaînement sortant, tous les enchaînements sont simultanément exécutés.  Pour un enchaînement entrant, tous les enchaînements ont été simultanément exécutés.	
<b>Ou inclusif</b> 	<b>Disjonction inclusive d'enchaînements.</b> Pour un enchaînement sortant, un ou plusieurs des enchaînements est exécuté.  Pour un enchaînement entrant, un ou plusieurs des enchaînements a été exécuté.	
<b>Ou exclusif</b> 	<b>Disjonction exclusive d'enchaînements.</b> Pour un enchaînement sortant, un des enchaînements est exécuté, à l'exclusion des autres.  Pour un enchaînement entrant, un des enchaînements a été exécuté, à l'exclusion des autres.	

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

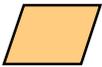
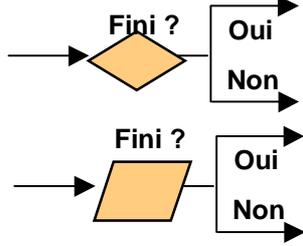
Symbole	Sémantique	Exemple
<b>Condition</b>   <b>Boucle</b> 	<b>Test.</b> Pour une alternative parmi une liste possible. Elle est souvent combinée à un algorithme au regard de son résultat.  <b>Itération.</b> Pour une itération bornée ou non. Elle est souvent combinée à un algorithme au regard de son résultat.	

Tableau 19 – Symboles de l'enchaînement des traitements

## 8.2 Traitements complexes

Cette section présente chaque traitement complexe du composant. Un traitement est dit complexe à partir du moment où il ne peut pas être compris par quiconque à la seule lecture du fichier source commenté.

Pour chaque traitement complexe il y a :

- La description de son algorithme.
- L'enchaînement des étapes de l'algorithme.  
Il se base sur :
  - L'organigramme.
  - La description des étapes critiques.
- Les particularités de l'algorithme.

Les symboles utilisés pour l'organigramme de chaque traitement complexe sont les mêmes que ceux présentés précédemment.

Voici un exemple pour l'algorithme **Chargement des contacts** :

	<b>Méthode de spécification technique d'un composant logiciel</b>	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
<b>Référence :</b> UpComp-Plan Qualité-000044-A Méthode de spécification technique d'un composant logiciel.doc		

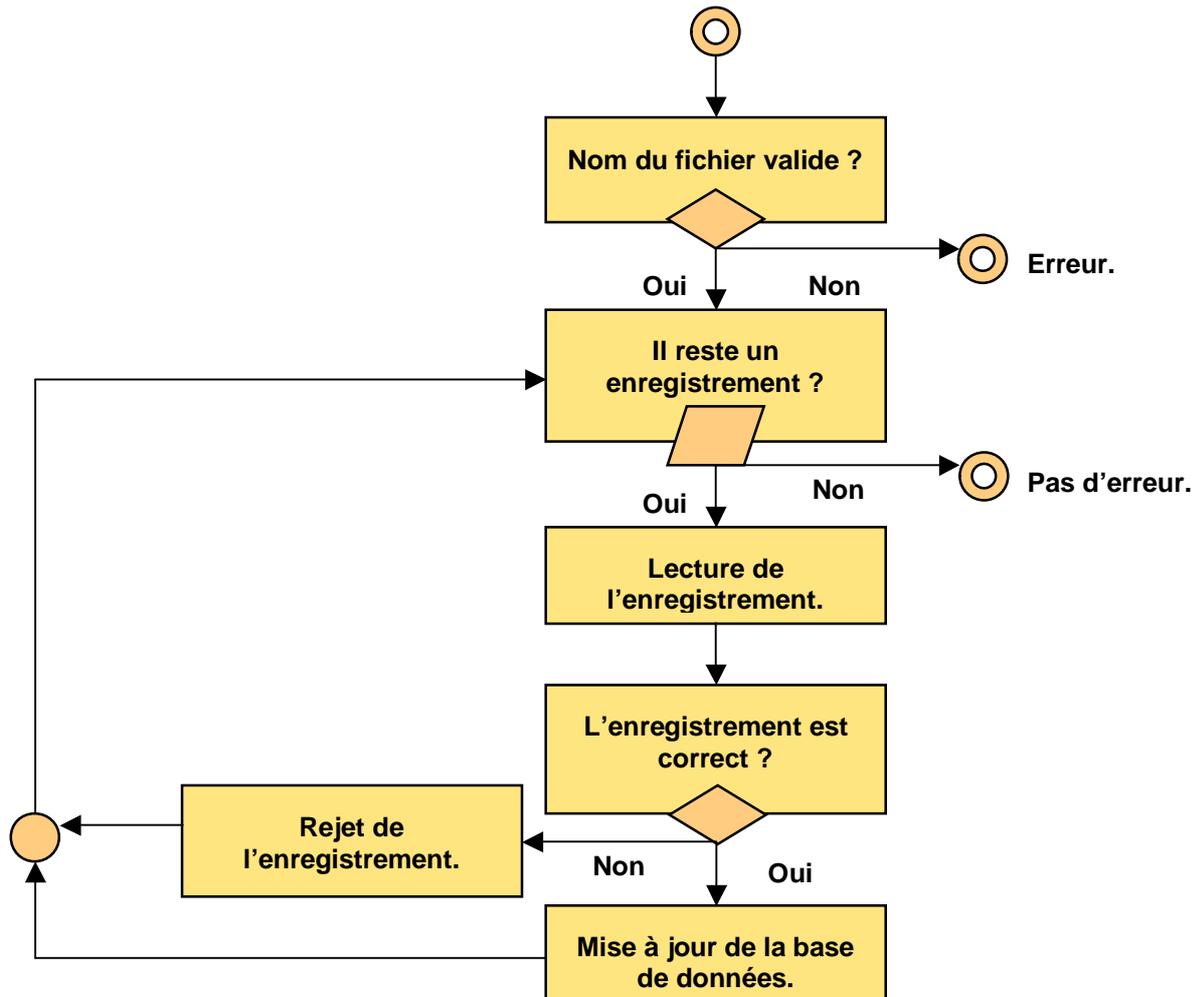


Diagramme 20 – Exemple d’algorithme d’un traitement complexe

Voici un exemple de description des étapes critiques :

- Si le fichier est inexistant, une erreur est inscrite dans le fichier journal.
- Si un enregistrement est incorrect en terme de syntaxe, une erreur est inscrite dans le journal et le programme s’arrête.
- Si un champ d’un enregistrement est incorrect, ce dernier est rejeté dans le fichier de rejet. Le programme continue.
- Si un champ d’un enregistrement ne respecte pas une règle de contrôle de validité, ce dernier est rejeté dans le fichier de rejet. Le programme continue.

Texte 21 – Exemple de description des étapes critiques d’un algorithme

**Fin de document**