

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

Suivi des versions-révisions et des validations du document.			
<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.

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

1 Tables

1.1 Table des matières

1	Tables.....	2
1.1	Table des matières.....	2
1.2	Table des illustrations.....	2
2	Références.....	3
2.1	Glossaire.....	3
2.2	Ressources.....	3
3	Introduction.....	4
3.1	Objet du document.....	4
3.2	Audience.....	4
3.3	Pré-requis.....	4
4	Interaction avec l'environnement.....	5
4.1	Description.....	5
4.2	Paramètres.....	5
4.3	Particularités.....	5
4.3.1	Compilation.....	5
4.3.2	Exécution.....	6
4.4	Application Program Interfaces.....	6
5	Choix techniques.....	8
6	Modèle de données.....	9
7	Composants du module.....	12

1.2 Table des illustrations

Tableau 1 – Exemple de déclaration d'un paramètre.....	5
Texte 2 – Exemple de déclaration des particularités de compilation.....	5
Texte 3 – Exemple de déclaration des particularités d'exécution.....	6
Texte 4 – Exemple de déclaration d'une API.....	7
Texte 5 – Exemple de déclaration d'un choix technique.....	8
Tableau 6 – Symboles du modèle physique de données.....	10
Diagramme 7 – Exemple de modèle physique de données.....	10
Tableau 8 – Exemple de description d'une entité.....	11
Tableau 9 – Exemple de composants du module.....	12

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.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.		
Un document est applicable 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.		
Un document est en référence 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.		
Un document applicable est indicé par A1, A2, A3 , etc. Un document en référence est indicé par R1, R2, R3 , etc.		
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-000043	Modèle de spécification technique d'un module logiciel.
R4	UpComp-Meilleure pratique-000017	Spécification technique d'un module.

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

3 Introduction

3.1 Objet du document

L'objet de ce document est de définir la méthode de spécification d'un module 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 :

- **Interaction avec l'environnement.**
Quel sont les paramètres ? Quelles sont les **Application Program Interfaces (API)** natives utilisées par ce module ? Quelles sont les particularités du module ?
- **Choix techniques.**
Quel sont les choix techniques retenus ? Pour quelles raisons ?
- **Modèle de données.**
Quel est le modèle physique des données ?
- **Composants techniques.**
Quels sont les composants techniques entrant dans la composition du module ?

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

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

Un exemple de l'application de la méthode est la meilleure pratique **Spécification technique d'un module** [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 module technique.
- **Ingénieurs de développement.**
Pour savoir comment est conçu le module technique.

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

3.3 Pré-requis

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

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

4 Interaction avec l'environnement

4.1 Description

Cette section présente globalement le module en faisant le lien entre :

- Sa **représentation externe**.
Elle décrite soit dans :
 - Les **Spécifications des besoins du système**.
 - Le **Guide de l'utilisateur**.
 - Le **Manuel d'administration**.
- Sa **réalité interne**.
Elle correspond au code des composants du module.

4.2 Paramètres

L'objet de cette section est de lister les paramètres du module logiciel en présentant pour chacun d'eux la liste des valeurs admissibles et un exemple d'emploi.

Voici un exemple pour le paramètre **Profil** :

Valeur du paramètre.	Sémantique.
Administratif.	Pour un profil administratif.
Directeur.	Pour un profil directeur.
Employé.	Pour un profil employé.

Tableau 1 – Exemple de déclaration d'un paramètre

Voici le contenu de chaque colonne de ce tableau :

- **Sémantique**.
Sémantique de la valeur admise par le paramètre.
 - **Valeur du paramètre**.
Valeur admise par le paramètre.
- S'il y a aucun paramètre, la mention « Néant » est inscrite.

4.3 Particularités

4.3.1 Compilation

Cette section décrit les particularités de compilation du module.

Voici un exemple :

- La définition **WIN32** spécifie de produire un exécutable pour **Windows**.
- La définition **LINUX20** spécifie de produire un exécutable pour **Linux**.

Texte 2 – Exemple de déclaration des particularités de compilation

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

S'il n'y a pas de particularité, la mention « Néant » est inscrite.

4.3.2 Exécution

Cette section décrit les particularités d'exécution du module.

Voici un exemple :

- La présence de la variable environnement **HTTP_HOME** spécifie que le module fonctionne en **Intranet**. Sinon, il fonctionne en client-serveur.

Texte 3 – Exemple de déclaration des particularités d'exécution

S'il n'y a pas de particularité, la mention « Néant » est inscrite.

4.4 Application Program Interfaces

Cette section présente les **Application Program Interfaces (API)** utilisées par les composants du module. Hormis **Up! Virtual technical Machine** et les pilotes techniques, il est rare qu'un module doit faire appel des **APIs** natives, puisque cela est contraire au principe de portabilité d'**Up! Application System**.

Chaque **API** est décrite de la manière suivante :

- **Prototypes.**
Liste des prototypes possibles de l'**API**.
La convention est d'écrire chaque mot-clé du langage en **gras et vert**.
- **Description.**
Description présentant l'utilité de l'**API**.
- **Exemple.**
Exemple d'emploi de l'**API**.
- **Avertissement.**
Lorsque l'**API** a un comportement singulier ou comporte un dysfonctionnement dans un usage particulier, celui-ci est mentionné.
- **Voir aussi.**
Renvoi vers d'autres concepts ou d'autres **APIs**.

Voici un exemple :



Méthode de spécification technique d'un module

Date rédaction :
17 novembre 2003.

Diffusion restreinte

Date validation :

Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc

4.4.1 _close – Win32

4.4.1.1 Prototypes

```
int _close(int NumeroFichier);
```

4.4.1.2 Description

Ferme le fichier identifié par *NumeroFichier*.

Le résultat est inutile.

4.4.1.3 Exemple

```
int NumeroFichier;
```

```
NumeroFichier=_open((LPCSTR)"c:\\tmp\\essai.txt", _O_BINARY);
```

```
...
```

```
_close(NumeroFichier);
```

4.4.1.4 Avertissement

Néant.

4.4.1.5 Voir aussi

`_open` pour ouvrir le fichier.

Texte 4 – Exemple de déclaration d'une API

Si aucune *Application Programme Interface (API)* n'est utilisée, la mention « Néant » est inscrite.

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

5 Choix techniques

Pour chaque sujet technique commun à l'ensemble des composants du module, il y a une section comportant :

- La description du sujet.
- Les différentes alternatives techniques possibles pour la mise en œuvre.
- Le choix qui a été retenu.
- Les raisons qui ont motivé ce choix.
- Les adaptations à porter au composant pour changer de choix.

Voici un exemple :

L'objet est la persistance de **Contacts** en terme de conservations des informations relatives aux adresses, aux contacts et aux sociétés, etc.

Les différentes alternatives possibles sont les suivantes :

- Utiliser une base de données relationnelle.
- Utiliser des fichiers d'enregistrements.
- Utiliser des fichiers à section.

Le choix retenu est la base de données relationnelle **My Sql** pour la raison suivante :

- **Disponibilité.**
Notre hébergeur nous propose une instance d'une base de données de ce type d'une taille suffisante.
- **Gratuité.**
Il n'y a pas de licence à payer et le coût d'exploitation est déjà compris dans le contrat d'hébergement.

En cas de changement de choix, les adaptations à apporter sont les suivantes :

- Pour changer de base de données relationnelle, il faut reprendre le code de lecture, d'écriture et de modification qui utilisent les **APIs** de **My Sql**.
- Pour les fichiers d'enregistrement, cela n'est d'aucune utilité, compte tenu de la puissance des bases de données relationnelles.

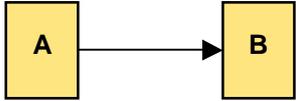
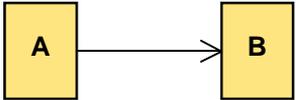
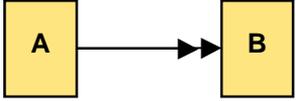
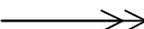
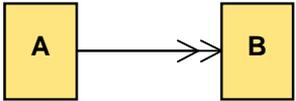
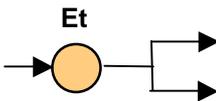
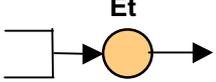
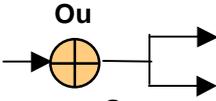
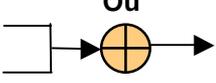
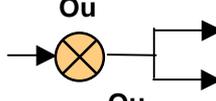
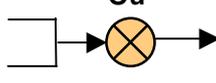
Texte 5 – Exemple de déclaration d'un choix technique

Si aucun choix technique n'a été effectué, la mention « Néant » est inscrite.

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

6 Modèle de données

Cette section présente le modèle physique de données commun à tous les composants du module. Les symboles suivants sont utilisés à cette fin :

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

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

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

Tableau 6 – Symboles du modèle physique de données

Voici un exemple :

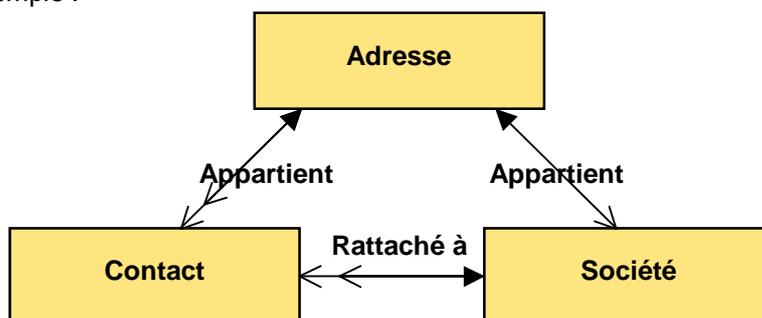


Diagramme 7 – Exemple de modèle physique de données

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 pour un fichier structuré par exemple, le diagramme du modèle est complété par la description complète de l'élément.

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

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

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

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

Tableau 8 – 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 description.**
La description permet de préciser la sémantique de la propriété.
 - **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 « »**.
- **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.

Si le module ne comporte aucun modèle de données, la mention « Néant » est inscrite.

	Méthode de spécification technique d'un module	Date rédaction : 17 novembre 2003.
	Diffusion restreinte	Date validation :
Référence : UpComp-Plan Qualité-000046-A Méthode de spécification technique d'un module.doc		

7 Composants du module

Cette section décrit les composants constituant le module.

Voici un exemple :

Fichiers du module.	
<ul style="list-style-type: none"> Fichier contacts.h – Importation des définitions. Fichier contacts.e – Définitions partagées par les composants dont le modèle de données. 	
Composants.	Contacts.
Description.	
Gestion des contacts : <ul style="list-style-type: none"> Ajout, modification, recherche et suppression d'une adresse. Ajout, modification, recherche et suppression d'un contact. Ajout, modification, recherche et suppression d'une société. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier contact.cpp – Code du composant. Fichier contact.h – Importation des définitions. Fichier contact.e – Définitions partagées par les composants dont le modèle de données. 	
Composants.	Demandes d'information.
Description.	
Gestion des demandes d'information : <ul style="list-style-type: none"> Ajout, modification, recherche et suppression d'une demande d'information. Réponse à une demande. Validation ou invalidation d'une demande. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier demande.cpp – Code du composant. Fichier demande.h – Importation des définitions. Fichier demande.e – Définitions partagées par les composants dont le modèle de données. 	

Tableau 9 – Exemple de composants du module

Voici le contenu de chaque colonne de ce tableau :

- Composants.**
Noms des composants constituant le module.
- Description.**
Description du composant.
- Fichiers.**
Noms des fichiers constituant le composant. Il s'agit de livrables tel un fichier source, un fichier d'exemple, une note annexe, etc.

Fin de document