

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.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	23 nov. 2003.	JPD.	Création.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

1 Tables

1.1 Table des matières

1	Tables	2
1.1	Table des matières	2
1.2	Table des illustrations	3
2	Références	4
2.1	Glossaire	4
2.2	Ressources	4
3	Introduction	5
3.1	Objet du document	5
3.2	Audience	5
3.3	Pré-requis	5
4	Interaction avec l'environnement	6
4.1	Description	6
4.2	Paramètres	6
4.3	Particularités	6
4.3.1	Compilation	6
4.3.2	Exécution	7
4.4	Application Program Interfaces	7
5	Choix techniques	8
5.1	Gestion des exceptions	8
5.2	Gestion des quotas	9
5.3	Gestion des contraintes	9
5.4	Gestion du dictionnaire de données et de traitements	9
5.5	Utilitaires pour Up ! Virtual Technical Machine	10
5.5.1	Création des objets standard	10
5.5.2	Conversions	13
5.5.3	Méthodes génériques	13
5.5.4	Calculs sur les scalaires	13
5.5.5	Manipulation des propriétés d'un objet	14
5.5.6	Manipulation des éléments d'un tableau	15
5.5.7	Manipulation des variables et des propriétés virtuelles	15
5.5.8	Interprétation d'un appel	15
5.5.9	Appel des méthodes au travers d'une interface	15
5.5.10	Vérification de l'intégrité des interfaces de traitements et des tables de méthodes	16
5.5.11	Exécution d'un appel dynamique	16
5.6	Trace de l'exécution	16
6	Modèle de données	18
7	Composants techniques	20

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

1.2 Table des illustrations

Texte 1 – Exemple de gestion des exceptions	8
Diagramme 2 – Modèle physique des données publiques du module <i>Up ! Kernel</i>	18
Tableau 3 – Glossaire du modèle physique des données publiques du module <i>Up ! Kernel</i>	19
Tableau 4 – Composants techniques du module	24

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.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 indiqué par A1, A2, A3 , etc. Un document en référence est indiqué 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-000046	Méthode de spécification technique d'un module.
A4	UpComp-UpsKrn-000002	Plan documentaire du projet.
A5	UpComp-UpsVm-000003	Plan de programmation.
A6	UpComp-UpsVm-000004	Programmation en C-- .
R7	http://www.up-comp.com	Site Internet d'Up ! Application System .
A8	UpComp-UpsKrn-000003	Plan d'écriture d'un module.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

3 Introduction

3.1 Objet du document

L'objet de ce document est de décrire le contenu technique du module logiciel **Up ! Kernel** pour le projet **Up ! Application System**.

Ce document est rédigé et approuvé par la **Maîtrise d'Oeuvre (MOE)**.

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 des documents suivants :

- **Méthode documentaire** [A1].
- **Processus de management de projet** [A2].
- **Méthode de spécification technique d'un module** [A3].

Nous rappelons que tous les documents applicables ou référencés pour le projet **Up ! Application System** sont tracés dans le **Plan documentaire** [A4].

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

4 Interaction avec l'environnement

4.1 Description

L'objet du module logiciel *Up ! Kernel* est de

- Mettre en oeuvre les objets de base d'*Up ! Virtual Technical Machine*.
Appel, ArbreBinaire, Binaire, Caractere, Date, Entier, Entrepot, Enumere, Exception, Liste, Module, Objet, Reel, Reference, Tableau, Type et *Transaction*.
- Gérer les exceptions.
Envoyer et attraper une exception. Libérer les ressources tant qu'une exception n'est pas attrapée.
- Gérer les quotas.
Pour les temps d'exécution uniquement. Les quotas sur le volume de données sont gérés par *Up ! Object Management System*.
- Gérer les contraintes.
Les contraintes sur les objets transactionnels sont différées par *Up ! Object Management System*.
- Gérer le dictionnaire de données et de traitements.
Pour les informations statiques via la bibliothèque dictionnaire de chaque module et pour les informations dynamiques.
- Fournir des utilitaires à *Up ! Virtual Technical Machine*.
Il y a :
 - Création des objets de base.
 - Trace de l'exécution.
 - Les méthodes et les co-méthodes génériques.
 - Les calculs sur les scalaires mettant en oeuvre les principes de sûreté de fonctionnement.
 - Conversion des scalaires vers les objets et réciproquement.
 - Manipulation des propriétés d'un objet.
 - Manipulation des éléments d'un tableau.
 - Manipulation des variables et des propriétés virtuelles.
 - Appel des méthodes au travers d'une interface.
 - Vérification de l'intégrité des interfaces de traitements et des tables de méthodes.
 - Exécution d'un appel dynamique.

4.2 Paramètres

Tous les paramètres sont documentés sur le [Site Internet d'Up ! Network](#) [R7].

4.3 Particularités

4.3.1 Compilation

Néant.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

4.3.2 Exécution

Néant.

4.4 Application Program Interfaces

Néant.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

5 Choix techniques

5.1 Gestion des exceptions

Pour envoyer une exception, il faut utiliser l'une des deux **Application Program Interfaces (API)** suivantes :

- **EnvoyerExceptionStandard** pour la tâche courante. Celle-ci est traitée immédiatement.
- **EnvoyerExceptionAUneTache** pour une autre tâche. Celle-ci est traitée en asynchrone dès qu'elle exécutera du code d'**Up ! Virtual Technical Machine**.

Pour attraper une exception, il faut procéder comme suit :

- Allouer un point de retour par l'appel à **UpsKrn.EmpilerPointDeRetour**.
- Se protéger par l'appel à **UpsVmSetJmp**.
 - En cas de code retour non nul, une exception a été reçue. Il est alors possible de :
 - Retrouver l'exception par l'appel à **UpsKrn.LireExceptionEnvoyee**.
 - Accepter l'exception par l'appel à **UpsKrn.LacherException**.
 - Annuler l'exception par l'appel à **UpsKrn.AnnulerExceptionEnAttente**.
 - Renvoyer l'exception par l'appel à **UpsKrn.RenvoyerException**.
 - En cas de code retour nul, le déroulement du programme s'effectue sur la section protégée. Avant de sortir de la section protégée, il faut libérer le point de retour par l'appel à **UpsKrn.DepilerPointDeRetour**.

Voici un exemple :

```

PointDeRetour=( *MonModIntTrtUpsKrn->EmpilerPointDeRetour) (Session);
if (UpsVmSetJmp( *PointDeRetour))
{
...
}
else
{
...
(*MonModIntTrtUpsKrn->DepilerPointDeRetour) (Session);
}

```

Texte 1 – Exemple de gestion des exceptions

Pour savoir si une exception a été envoyée et qu'elle n'a pas encore été prise en compte, il faut appeler l'**Application Program Interface (API)** **UpsKrn.LireExceptionEnvoyee**.

Une attente active peut alors demander à traiter les exceptions en attente par l'appel à l'**Application Program Interface (API)** **UpsKrn.TraiterExceptionEnAttente**.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

5.2 Gestion des quotas

Pour poser l'audit des quotas sur le temps d'exécution en début d'un traitement, il faut utiliser l'une des deux **Application Program Interfaces (API)** suivantes :

- **UpsKrn.EmplierQuotaAppel.**
Pour une procédure ou une fonction.
- **UpsKrn.EmplierQuotaMethode.**
Pour une méthode.

Pour supprimer l'audit des quotas sur le temps d'exécution en fin d'un traitement, il faut utiliser l'**Application Program Interfaces (API)** **UpsKrn.DepilerQuota.**

5.3 Gestion des contraintes

Les **Application Program Interfaces (API)** suivantes permettent gérer les contraintes :

- **UpsKrn.VerifierContrainteNonNul.**
Vérifie qu'un objet est non nul.
- **UpsKrn.VerifierContrainteSelon.**
Vérifie que la partie polymorphe cible d'un type est accessible.
- **UpsKrn.VerifierContrainteEntier.**
Vérifie une contrainte sur le type **Entier**.
- **UpsKrn.VerifierContrainteReel.**
Vérifie une contrainte sur le type **Reel**.
- **UpsKrn.VerifierContrainteObjet.**
Vérifie une contrainte sur le type **Objet**.
- **UpsKrn.TenterDeValiderContraintes.**
Tente de valider les contraintes gelées pour savoir si une transaction peut être validée. Le résultat est **Vrai** en cas de succès.
- **UpsKrn.LibererContraintes.**
Annule les contraintes en attente lors de l'invalidation d'une transaction.

5.4 Gestion du dictionnaire de données et de traitements

Les **Application Program Interface (API)** pour la gestion du dictionnaire de données et de traitements sont les suivantes :

- **UpsKrn.MethodeLirePropriete** et **UpsKrn.CoMethodeLirePropriete.**
Fonction LirePropriete(NomPropriete : Caractere) **Retourner Nul Ou** Objet;
- **UpsKrn.MethodeEcrirePropriete** et **UpsKrn.CoMethodeEcrirePropriete.**
Procedure EcrirePropriete(NomPropriete : Caractere, ValeurPropriete : **Nul Ou** Objet);
- **UpsKrn.MethodeIncrementerDecrementerPropriete** et **UpsKrn.CoMethodeIncrementerDecrementerPropriete.**
Fonction IncrementerDecrementerPropriete(NomPropriete : **Nul Ou** Caractere, Incrementer : Booleen, Prefixe : Booleen) **Retourner Nul Ou** Objet;
- **UpsKrn.MethodeSupprimerPropriete** et **UpsKrn.CoMethodeSupprimerPropriete.**
Procedure SupprimerPropriete(NomPropriete : Caractere);

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

- ***UpsKrn.MethodeEnumererProprietes*** et ***UpsKrn.CoMethodeEnumererProprietes***.
Fonction EnumererProprietes(Numero : Entier, TypePropriete : **Nul Ou** Type **Sortie**, EstNul : Booleen **Sortie**, Aide : **Nul Ou** Caractere **Sortie**, NomEnumere : **Nul Ou** Caractere **Sortie**) **Retourner** **Nul Ou** Caractere;

a Les propriétés dynamiques sont gérées sous forme d'une liste d'associations (NomDeLaPropriété, Objet).

5.5 Utilitaires pour Up ! Virtual Technical Machine

5.5.1 Création des objets standard

Les **Application Program Interfaces (API)** suivantes permettent de créer aisément les objets standards d'**Up ! Kernel** :

- ***UpsKrn.CreerObjelAppel***.
Création d'un objet de type **Appel** à partir de son appel et son co-appel.
- ***UpsKrn.CreerObjetBinaire***.
Création d'un objet de type **Binaire** à partir de sa valeur.
- ***UpsKrn.CreerObjetCaractere***.
Création d'un objet de type **Caractere** à partir de sa valeur.
- ***UpsKrn.CreerObjetDate***.
Création d'un objet de type **Date** initialisée au 1^{er} janvier de l'an 1 à 0 heure, 0 minute et 0 seconde.
- ***UpsKrn.CreerObjetEntier***.
Création d'un objet de type **Entier** à partir de sa valeur.
- ***UpsKrn.CreerObjetEntrepot***.
Création d'un objet de type **Entrepot**.
- ***UpsKrn.CreerObjetEnumere***.
Création d'un objet de type **Enumere** à partir de sa valeur.
- ***UpsKrn.CreerObjetException***.
Création d'un objet de type **Exception** à partir de ses propriétés.
- ***UpsKrn.CreerObjetListe***.
Création d'un objet de type **Liste** contenant un objet.
- ***UpsKrn.CreerObjetModule***.
Création d'un objet de type **Module**.
- ***UpsKrn.CreerObjetReel***.
Création d'un objet de type **Reel** à partir de sa valeur.
- ***UpsKrn.CreerObjetReference***.
Création d'un objet de type **Reference** pointant un objet.
- ***UpsKrn.CreerObjetTableau***.
Création d'un objet de type **Tableau** contenant un objet.

5.5.1.1 Objets de type Appel

Un objet de type **Appel** encapsule une ressource exécutable dont le type est défini par l'énuméré **EnuUpsVmTypeDeTraitements** :

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

- **TT_NatifCompile.**
Il s'agit de code compilé en **C--** écrit manuellement ou généré par **Up ! Compiler**.
- **TT_ScriptInterprete.**
Il s'agit de code interprété en **Up ! 5GL** interprété par **Up ! Engine**.
- **TT_NatifCorba.**
Il s'agit de code compilé en **C++** écrit manuellement ou généré par **Up ! Compiler** dont l'interface d'appel répond à la norme **Common Object Request Broker Architecture**.
- **TT_NatifDCom.**
Il s'agit de code compilé en **C++** écrit manuellement ou généré par **Up ! Compiler** dont l'interface d'appel répond à la norme **Component Object Module**.
- **TT_NatifJava.**
Il s'agit de code compilé en **Java** écrit manuellement ou généré par **Up ! Compiler** dont l'interface d'appel répond à la norme **Java Native Interface**.

Pour le type autre que natif compilé, chaque objet appel comporte une extension permettant de conserver les informations propriétaires nécessaires au cycle de vie de l'appel dans la technologie tierce. Cette extension est gérée au travers d'une table de méthodes spéciales de type **TypUpsKrnGestionAppelTiers** qu'il est nécessaire de déclarer avec l'**Application Program Interface (API) ReferencerGestionAppelTiers** dans les modules gestionnaires spécialisés sur cette technologie.

Voici les méthodes attendues :

- **CoAppel.**
Enumération les paramètres de l'appel.
- **Destructeur.**
Destruction des informations propriétaires de l'appel.
- **MethodeComparerAppels.**
Retourne **Vrai** si deux appels sont identiques en terme de foncteur et de prototype.
- **MethodeCloner.**
Clonage des informations propriétaires d'un appel.
- **MethodeExporter.**
Exportation des informations propriétaires d'un appel.
- **MethodImporter.**
Importation des informations propriétaires d'un appel.
- **ExecuterProcedureSansParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une procédure sans paramètre.
- **ExecuterProcedureAvecParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une procédure avec paramètre.
- **ExecuterFonctionEnumereCourtSansParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un énuméré court sans paramètre.
- **ExecuterFonctionEnumereCourtAvecParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un énuméré court avec paramètre.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

- **ExecuterFonctionEnumereLongSansParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un énuméré long sans paramètre.
- **ExecuterFonctionEnumereLongAvecParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un énuméré long avec paramètre.
- **ExecuterFonctionEntierSansParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un entier sans paramètre.
- **ExecuterFonctionEntierAvecParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un entier avec paramètre.
- **ExecuterFonctionReelSansParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un réel sans paramètre.
- **ExecuterFonctionReelAvecParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un réel avec paramètre.
- **ExecuterFonctionObjetSansParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un objet sans paramètre.
- **ExecuterFonctionObjetAvecParametre.**
Exécution de l'appel en technologie tierce. Il doit correspondre à une fonction retournant un objet avec paramètre.

5.5.1.2 Objets de type Caractere ou Binaire

Pour retrouver la valeur d'un objet de type **Caractere** ou **Binaire**, il faut utiliser **Application Program Interfaces (API) UpsKrn.CalculerBuffer**.

Pour retrouver la taille de la valeur d'un objet de type **Caractere** ou **Binaire**, il faut utiliser **Application Program Interfaces (API) UpsKrn.CalculerTaille**.

M Normalement, la valeur d'un objet de type **Caractere** ou **Binaire** n'est jamais modifiée. Il faut en allouer un autre qui contiendra la valeur modifiée.

5.5.1.3 Objets de type Enumere

Les calculs sur les objets de type **Enumere** sont les suivants :

- **UpsKrn.BooleenEt.**
Opérateur Et(B : Nul Ou Booleen) Retourner Nul Ou Booleen;
- **UpsKrn.BooleenOu.**
Opérateur Ou(B : Nul Ou Booleen) Retourner Nul Ou Booleen;
- **UpsKrn.BooleenOuExclusif.**
Opérateur OuExclusif(B : Nul Ou Booleen) Retourner Nul Ou Booleen;
- **UpsKrn.BooleenNon.**
Opérateur Non() Retourner Nul Ou Booleen;

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

5.5.2 Conversions

Les **Application Program Interface (API)** pour les conversions sont les suivantes :

- **UpsKrn.ExtraireEnumereCourt.**
Pour convertir un objet d'un type **Enumere** en **TypUpsVmUnsignedChar**.
- **UpsKrn.ExtraireEnumereLong.**
Pour convertir un objet d'un type **Enumere** en **TypUpsVmUnsignedShort**.
- **UpsKrn.ExtraireEntier.**
Pour convertir un objet d'un type **Entier** en **TypUpsVmLong**.
- **UpsKrn.ExtraireReel.**
Pour convertir un objet d'un type **Reel** en **TypUpsVmDouble**.
- **UpsKrn.VerifierOrigineObjet.**
Pour convertir un objet d'un type dans un autre.

5.5.3 Méthodes génériques

Les méthodes génériques sont les suivantes :

- **UpsKrn.MethodeAffecter** et **UpsKrn.CoMethodeAffecter.**
Operateur =(P : **Null** Ou **Objet**);
- **UpsKrn.MethodeEgal** et **UpsKrn.CoMethodeEgal.**
Operateur ==(P : **Null** Ou **Objet**) **Retourner Null** Ou **Booleen**;
- **UpsKrn.MethodeDifferent** et **UpsKrn.CoMethodeDifferent..**
Operateur !=(P : **Null** Ou **Objet**) **Retourner Null** Ou **Booleen**;
- **UpsKrn.MethodeComposer** et **UpsKrn.CoMethodeComposer.**
Operateur ?=(P : **Null** Ou **Objet**);
- **UpsKrn.ClonerObjet** et **UpsKrn.CoMethodeCloner.**
Fonction Cloner (EntrepotCible : **Null** Ou **Entrepot=Null**, Profondeur : **ProfondeurClonage=ClonageObjetSeul**) **Retourner Null** Ou **Objet**;
- Autres co-méthodes standards.

5.5.4 Calculs sur les scalaires

5.5.4.1 Pour les entiers

Les calculs sur les scalaires entiers intégrant les normes de sûreté de fonctionnement sont les suivants :

- **UpsKrn.DiviserEntier.**
- **UpsKrn.ModulerEntier.**
- **UpsKrn.MiseAJourVariableEntier.**
Pour appliquer un opérateur.
- **UpsKrn.IncrementerDecrementerVariableEntier.**

5.5.4.2 Pour les réels

Les calculs sur les scalaires réels intégrant les normes de sûreté de fonctionnement sont les suivants :

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

- ***UpsKrn.DiviserReel.***
- ***UpsKrn.ModulerReel.***
- ***UpsKrn.MiseAJourVariableReel.***
Pour appliquer un opérateur.
- ***UpsKrn.IncrementerDecrementerVariableReel.***
- ***UpsKrn.ComparerReels.***
Compare deux nombres réels en tenant compte des arrondis de calcul.

5.5.5 Manipulation des propriétés d'un objet

5.5.5.1 Lecture des propriétés des objets

Les ***Application Program Interface (API)*** pour la lecture des propriétés d'un objet sont les suivantes :

- ***UpsKrn.ExtraireChampEnumereCourt.***
- ***UpsKrn.ExtraireChampEnumereLong.***
- ***UpsKrn.ExtraireChampEntier.***
- ***UpsKrn.ExtraireChampReel.***
- ***UpsKrn.ExtraireChampObjet.***
- ***UpsKrn.ExtraireInstanceDe.***
Pour retrouver le type d'un objet.

5.5.5.2 Ecriture des propriétés des objets

Les ***Application Program Interface (API)*** pour l'écriture des propriétés d'un objet sont les suivantes :

- ***UpsKrn.MiseAJourChampEnumereCourt.***
- ***UpsKrn.MiseAJourChampEnumereLong.***
- ***UpsKrn.MiseAJourChampEntier.***
- ***UpsKrn.MiseAJourChampReel.***
- ***UpsKrn.MiseAJourChampObjet.***

5.5.5.3 Destruction des propriétés des objets

Les ***Application Program Interface (API)*** pour la destructions des propriétés d'un objet sont les suivantes :

- ***UpsKrn.DetruireChampObjet.***
- ***UpsKrn.MiseAJourChampNul.***

5.5.5.4 Opérations sur les propriétés des objets

Les ***Application Program Interface (API)*** pour les opérations sur les propriétés d'un objet sont les suivantes :

- ***UpsKrn.MethodeComposerChampObjet.***
- ***UpsKrn.IncrementerDecrementerChampEntier.***

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

- *UpsKrn.IncrementerDecrementerChampReel.*
- *UpsKrn. IncrementerDecrementerChampObjet.*

5.5.6 Manipulation des éléments d'un tableau

Les *Application Program Interface (API)* pour la manipulation des éléments d'un tableau sont les suivantes :

- *UpsKrn.MiseAJourElementTableau.*
- *UpsKrn.MethodeComposerElementTableau.*
- *UpsKrn.IncrementerDecrementerChampReel.*
- *UpsKrn.IncrementerDecrementerElementTableau.*

5.5.7 Manipulation des variables et des propriétés virtuelles

5.5.7.1 Variables virtuelles

Les *Application Program Interface (API)* pour la manipulation variables virtuelles sont les suivantes :

- *UpsKrn.VirtuelVariable\$NomType\$.*
Par exemple *UpsKrn.VirtuelVariableEntier.*
- *UpsKrn.IncrementerDecrementerVirtuelVariable\$NomType\$.*
Par exemple *UpsKrn.VirtuelVariableEntie..*

5.5.7.2 Propriétés virtuelles

Les *Application Program Interface (API)* pour la manipulation propriétés virtuelles sont les suivantes :

- *UpsKrn.VirtuelChamp\$NomType\$.*
Par exemple *UpsKrn.VirtuelChampEntier.*
- *UpsKrn.IncrementerDecrementerVirtuelChamp\$NomType\$.*
Par exemple *UpsKrn.IncrementerDecrementerVirtuelChampEntier.*

5.5.8 Interprétation d'un appel

Les *Application Program Interface (API)* pour le déclenchement de la ressource exécutable d'un appel sont les suivantes :

- *UpsKrn.\$NomDeLAppel\$.*
Par exemple *UpsKrn.ProcedureSansParametre.*

5.5.9 Appel des méthodes au travers d'une interface

Les *Application Program Interface (API)* pour l'appel des méthodes au travers d'une interface sont les suivantes :

- *UpsKrn.I\$NomDeLaMethode\$.*
Par exemple *UpsKrn.IConstructeur.*

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

5.5.10 Vérification de l'intégrité des interfaces de traitements et des tables de méthodes

Les **Application Program Interface (API)** pour la vérification de l'intégrité des interfaces de traitements et des tables de méthodes sont les suivantes :

- **UpsKrn.VerifierInterfaceTraitements.**
Pour une interface de traitements d'une bibliothèque ne correspondant pas à un adaptateur d'un logiciel de base.
- **UpsKrn.VerifierInterfaceTraitementsNat.**
Pour une interface de traitements d'une bibliothèque correspondant à un adaptateur d'un logiciel de base.
- **UpsKrn.VerifierInterfaceMethodes.**
Pour une table des méthodes.

M Hormis dans l'en-tête des méthodes de type **TypUpsVmEnteteMethodes** ou de type **TypUpsVmEnteteMethodesModules**, aucune méthode ne doit avoir la valeur **NULL**. Elles doivent être renseignées à défaut à **UpsKrn.AppellImpossibleApiNull**.

5.5.11 Exécution d'un appel dynamique

Les **Application Program Interface (API)** pour l'exécution d'un appel dynamique sont les suivantes :

- **UpsKrn.\$NomDuTypeDuPrototype\$.**
Par exemple **UpsKrn.ProcedureSansParametre.**

5.6 Trace de l'exécution

La trace de l'exécution d'un programme est écrite dans un flux adossée à un fichier dès lors qu'elle est activée.

Une trace est écrite au moyen de l'**Application Program Interface (API)** **UpsKrn.MiseAuPoint** utilisant l'énuméré **EnuUpsKrnTypeTrace** servant à coder la nature de la trace à écrire :

- **TT_Appel.**
Trace l'entrée dans un nouvel appel de procédure, de fonction ou de méthode.
- **TT_ParametreBooleen.**
Trace la transmission d'un paramètre correspondant à un booléen en entrée ou en sortie.
- **TT_ParametreEntier.**
Trace la transmission d'un paramètre correspondant à un nombre entier en entrée ou en sortie.
- **TT_ParametreEnumere.**
Trace la transmission d'un paramètre correspondant à un énuméré en entrée ou en sortie.
- **TT_ParametreReel.**
Trace la transmission d'un paramètre correspondant à un nombre réel en entrée ou en sortie.
- **TT_ParametreObjet.**
Trace la transmission d'un paramètre correspondant à un objet en entrée ou en sortie.
- **TT_Retourner.**
Trace la sortie d'une procédure ou d'une méthode procédurale.
- **TT_RetournerBooleen.**
Trace la sortie d'une fonction ou d'une méthode fonctionnelle retournant un booléen.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

- **TT_RetournerEntier.**
Trace la sortie d'une fonction ou d'une méthode fonctionnelle retournant un nombre entier.
- **TT_RetournerEnumere.**
Trace la sortie d'une fonction ou d'une méthode fonctionnelle retournant un énuméré.
- **TT_RetournerReel.**
Trace la sortie d'une fonction ou d'une méthode fonctionnelle retournant un nombre réel.
- **TT_RetournerObjet.**
Trace la sortie d'une fonction ou d'une méthode fonctionnelle retournant un objet.
- **TT_Exception.**
Trace l'envoi d'une exception.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

6 Modèle de données

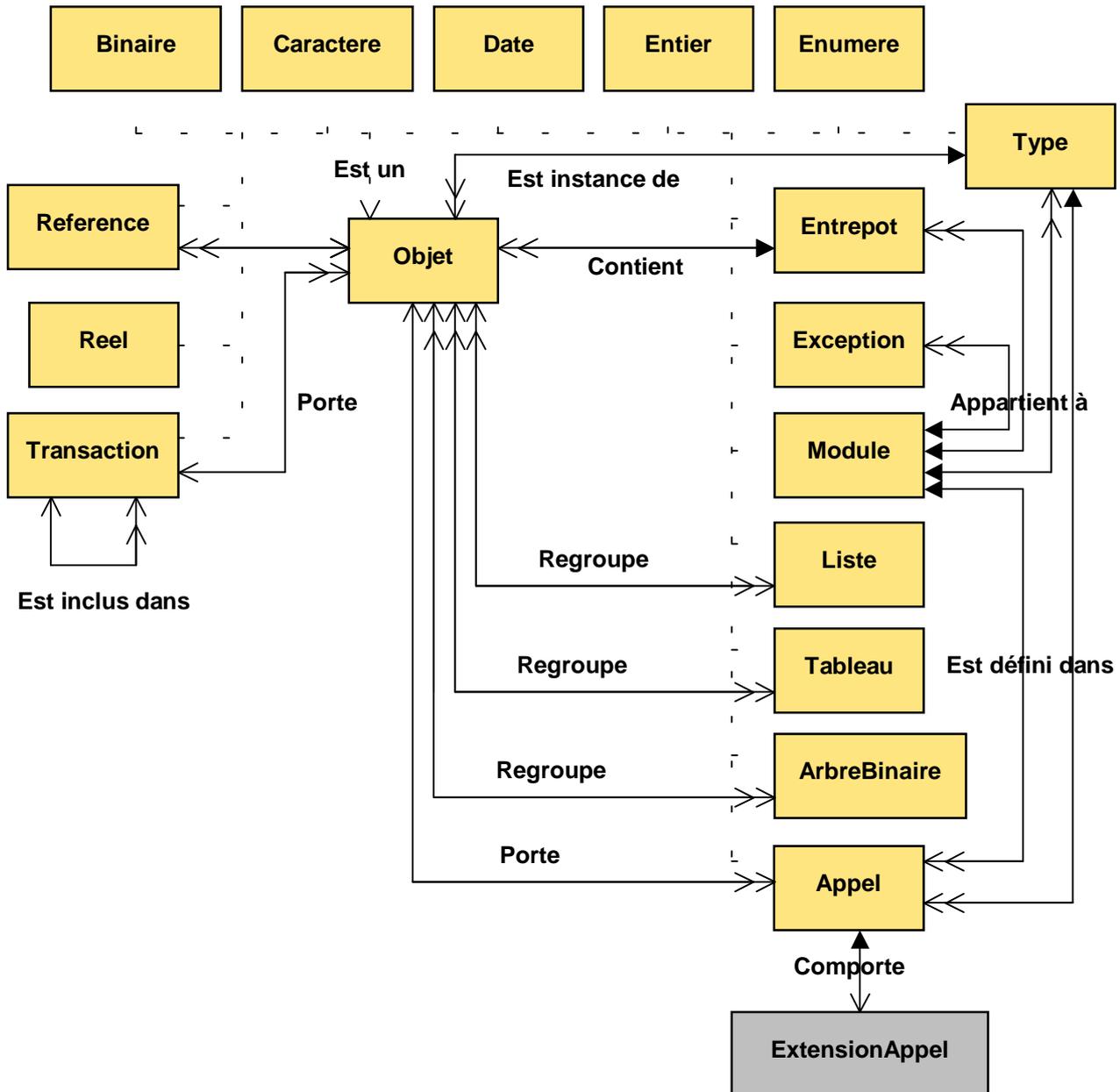


Diagramme 2 – Modèle physique des données publiques du module Up ! Kernel

Toutes les entités suivantes sont décrites dans le fichier **upskrn.e** :

Entité.	Description.
Appel.	Modélisation d'un objet appel.
ArbreBinaire.	Modélisation d'un objet arbre binaire.
Binaire.	Modélisation d'un objet binaire.
Caractere.	Modélisation d'un objet caractère.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

Date.	Modélisation d'un objet date.
Entier.	Modélisation d'un objet entier.
Entrepot.	Modélisation d'un objet entrepôt.
Enumere.	Modélisation d'un objet énuméré.
Exception.	Modélisation d'un objet exception.
ExtensionAppel.	Extension d'appel pour sa modélisation dans une technologie tierce.
Liste.	Modélisation d'un objet liste.
Module	Modélisation d'un objet module.
Objet.	Modélisation d'un objet générique.
Reel.	Modélisation d'un objet réel.
Reference.	Modélisation d'un objet référence.
Tableau.	Modélisation d'un objet tableau.
Transaction.	Modélisation d'un objet transaction.
Type.	Modélisation d'un objet type.

Tableau 3 – Glossaire du modèle physique des données publiques du module *Up ! Kernel*

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

7 Composants techniques

Le module **Up ! Kernel** pour le projet **Up ! Application System** est constitué des composants suivants :

Fichiers du module.	
<ul style="list-style-type: none"> Fichier upskrn.e – Définition des types gérés par Up ! Kernel. Fichier upskrn.h – En-tête privé de Up ! Kernel. Fichier upskrn.def – Exportation des symboles pour Windows. 	
Composants.	upskrn0.
Description.	
Interface entre Up ! Kernel et Up ! Module .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn0.cpp – Fichier source. Fichier upskrn0.h – En-tête privé de upskrn0.cpp. Fichier upskrn0.e – En-tête protégé de upskrn0.cpp. 	
Composants.	upskrn1.
Description.	
<ul style="list-style-type: none"> Gestion de BinaireOuCaractere. Méthodes du type Binaire. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn1.cpp – Fichier source. Fichier upskrn1.h – En-tête privé de upskrn1.cpp. Fichier upskrn1.e – En-tête protégé de upskrn1.cpp. 	
Composants.	upskrn2.
Description.	
Méthodes du type Caractere .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn2.cpp – Fichier source. Fichier upskrn2.h – En-tête privé de upskrn2.cpp. Fichier upskrn2.e – En-tête protégé de upskrn2.cpp. 	
Composants.	upskrn3.
Description.	
Méthodes du type Date .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn3.cpp – Fichier source. Fichier upskrn3.h – En-tête privé de upskrn3.cpp. Fichier upskrn3.e – En-tête protégé de upskrn3.cpp. 	
Composants.	upskrn4.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

Description.	
Méthodes du type Entier .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn4.cpp – Fichier source. Fichier upskrn4.h – En-tête privé de upskrn4.cpp. Fichier upskrn4.e – En-tête protégé de upskrn4.cpp. 	
Composants.	upskrn5.
Description.	
<ul style="list-style-type: none"> Méthodes du type Enumere. Calculs sur les entiers. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn5.cpp – Fichier source. Fichier upskrn5.h – En-tête privé de upskrn5.cpp. Fichier upskrn5.e – En-tête protégé de upskrn5.cpp. 	
Composants.	upskrn6.
Description.	
<ul style="list-style-type: none"> Méthodes du type Exception.. Gestion des exceptions. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn6.cpp – Fichier source. Fichier upskrn6.h – En-tête privé de upskrn6.cpp. Fichier upskrn6.e – En-tête protégé de upskrn6.cpp. 	
Composants.	upskrn7.
Description.	
<ul style="list-style-type: none"> Méthodes du type Objet. Méthodes et co-méthodes génériques. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn7.cpp – Fichier source. Fichier upskrn7.h – En-tête privé de upskrn7.cpp. Fichier upskrn7.e – En-tête protégé de upskrn7.cpp. 	
Composants.	upskrn8.
Description.	
<ul style="list-style-type: none"> Méthodes du type Reel. Calculs sur les réels. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn8.cpp – Fichier source. Fichier upskrn8.h – En-tête privé de upskrn8.cpp. Fichier upskrn8.e – En-tête protégé de upskrn8.cpp. 	
Composants.	upskrn9.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

Description.	
<ul style="list-style-type: none"> Méthodes du type Type. Manipulation des champs des objets. Conversion des scalaires en objet et réciproquement. Gestion des contraintes. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn9.cpp – Fichier source. Fichier upskrn9.h – En-tête privé de upskrn9.cpp. Fichier upskrn9.e – En-tête protégé de upskrn9.cpp. 	
Composants.	upskrn10.
Description.	
Méthodes du type Liste .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn10.cpp – Fichier source. Fichier upskrn10.h – En-tête privé de upskrn10.cpp. Fichier upskrn10.e – En-tête protégé de upskrn10.cpp. 	
Composants.	upskrn11.
Description.	
Méthodes du type Reference .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn11.cpp – Fichier source. Fichier upskrn11.h – En-tête privé de upskrn11.cpp. Fichier upskrn11.e – En-tête protégé de upskrn11.cpp. 	
Composants.	upskrn12.
Description.	
<ul style="list-style-type: none"> Méthodes du type Tableau. Manipulation des éléments d'un tableau. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn12.cpp – Fichier source. Fichier upskrn12.h – En-tête privé de upskrn12.cpp. Fichier upskrn12.e – En-tête protégé de upskrn12.cpp. 	
Composants.	upskrn13.
Description.	
Méthodes du type ArbreBinaire .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn13.cpp – Fichier source. Fichier upskrn13.h – En-tête privé de upskrn13.cpp. Fichier upskrn13.e – En-tête protégé de upskrn13.cpp. 	
Composants.	upskrn14.

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

Description.	
Méthodes du type Entrepôt .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn14.cpp – Fichier source. Fichier upskrn14.h – En-tête privé de upskrn14.cpp. Fichier upskrn14.e – En-tête protégé de upskrn14.cpp. 	
Composants.	upskrn15.
Description.	
Méthodes du type Module .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn15.cpp – Fichier source. Fichier upskrn15.h – En-tête privé de upskrn15.cpp. Fichier upskrn15.e – En-tête protégé de upskrn15.cpp. 	
Composants.	upskrn16.
Description.	
<ul style="list-style-type: none"> Méthodes du type Appel. Vérification de l'intégrité des interfaces de traitements et les tables de méthodes. Exécution d'appels dynamiques. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn16.cpp – Fichier source. Fichier upskrn16.h – En-tête privé de upskrn16.cpp. Fichier upskrn16.e – En-tête protégé de upskrn16.cpp. 	
Composants.	upskrn17.
Description.	
Méthodes du type Transaction .	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn17.cpp – Fichier source. Fichier upskrn17.h – En-tête privé de upskrn17.cpp. Fichier upskrn17.e – En-tête protégé de upskrn17.cpp. 	
Composants.	upskrn99.
Description.	
<ul style="list-style-type: none"> Interface entre Up ! Kernel et lui-même. Gestion des bibliothèques dictionnaires. Gestion des variables virtuelles et des champs virtuels. Gestion des méthodes appelées au travers d'une interface. 	
Fichiers.	
<ul style="list-style-type: none"> Fichier upskrn99.cpp – Fichier source. Fichier upskrn99.h – En-tête privé de upskrn99.cpp. Fichier upskrn99.e – En-tête protégé de upskrn99.cpp. 	

	Spécification technique du module	Date rédaction : 12 mai 2004.
	Diffusion restreinte	Date validation :
Référence : UpComp-UpsKrn-000004-A Spécification technique du module UpsKrn.doc		

Tableau 4 – Composants techniques du module

Fin de document