# Crésus Core : redesign majeur en cours

# Personne physique et contacts

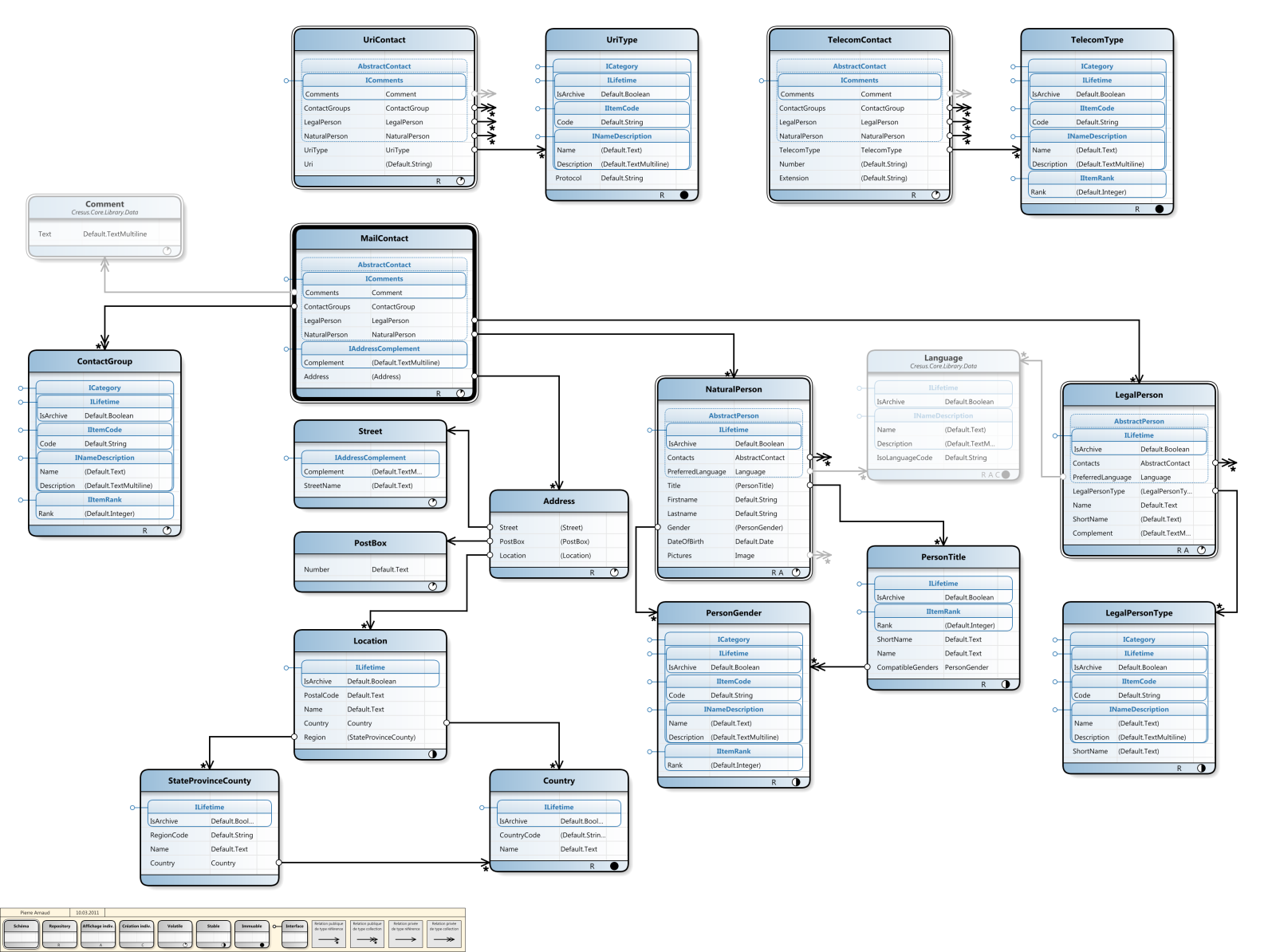


Schéma des entités, tel que dessiné avec Designer 3.2.

# Définition de l’interface graphique

L’interface graphique est construite par des contrôleurs. Jusqu’à peu, ces contrôleurs devaient être décrits avec beaucoup de code C# pour définir les tuiles qui composent les colonnes de l’interface.

Nous souhaitons simplifier ce codage et je propose le recours à une programmation de type *fluent interface* :

this.AddBrick ()

.Name ("NaturalPerson")

.Icon ("Data.NaturalPerson")

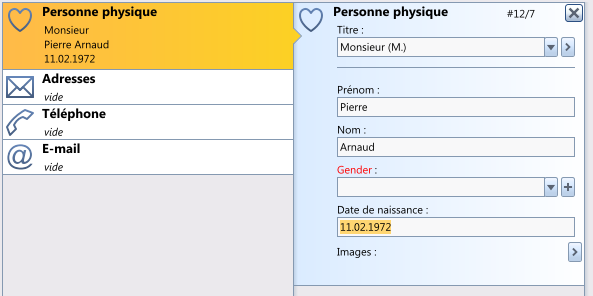
.Title (TextFormatter.FormatText ("Personne physique"))

.TitleCompact (TextFormatter.FormatText ("Personne"))

.Text (x => x.GetSummary ())

.TextCompact (x => x.GetCompactSummary ())

.End ();

this.AddBrick (x => x.Contacts)

.AsType<MailContactEntity> ()

.Name ("MailContact")

.Icon ("Data.MailContact")

.Title ("Adresses")

.TitleCompact ("Adresses")

.Text (CollectionTemplate.DefaultEmptyText)

.Template ()

.Title (x => x.GetTitle ())

.Text (x => x.GetSummary ())

.TextCompact (x => x.GetCompactSummary ())

.End ();

this.AddBrick (x => x.Contacts)

.AsType<TelecomContactEntity> ()

.AutoGroup ()

.Name ("TelecomContact")

.Icon ("Data.TelecomContact")

.Title ("Téléphones")

.TitleCompact ("Téléphones")

.Text (CollectionTemplate.DefaultEmptyText)

.Template ()

.Text (x => x.GetSummary ())

.TextCompact (x => x.GetCompactSummary ())

.End ();

this.AddBrick (x => x.Contacts)

.AsType<UriContactEntity> ()

.AutoGroup ()

.Name ("UriContact")

.Icon ("Data.UriContact")

.Title ("E-mails")

.TitleCompact ("E-mails")

.Text (CollectionTemplate.DefaultEmptyText)

.Template ()

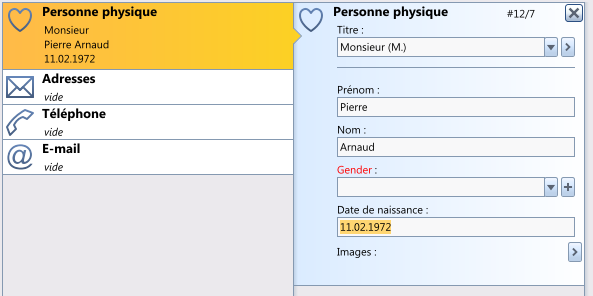
.Text (x => x.GetSummary ())

.TextCompact (x => x.GetCompactSummary ())

.End ();

# Simplifications supplémentaires

Afin d’éviter de devoir répéter constamment des éléments qui peuvent être déduits par le programme, la valeur des propriétés de l’interface est déduite en fonction du contexte (inspiré des principes *DRY Don’t Repeat Yourself* et *Convention over Configuration*).



Voici ce que cela pourrait donner :

this.AddBrick ();

this.AddBrick (x => x.Contacts)

.AsType<MailContactEntity> ()

.Template (title: x => x.GetTitle ());

this.AddBrick (x => x.Contacts)

.AsType<TelecomContactEntity> ()

.AutoGroup ();

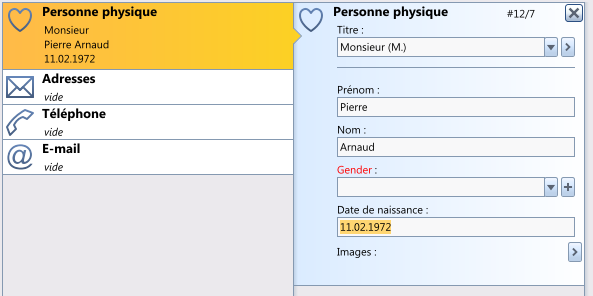
this.AddBrick (x => x.Contacts)

.AsType<UriContactEntity> ()

.AutoGroup ();

# Câblage des masques de saisie

Les masques de saisie peuvent aussi être définis au moyen des mêmes constructions, en utilisant des propriétés Input qui spécifient les divers champs avec des propriétés Field.



this.AddBrick ()

.Input ()

.Field (x => x.Title)

.End ()

.Input ()

.Field (x => x.Firstname)

.Field (x => x.Lastname)

.End ()

.Separator ()

.Input ()

.Field (x => x.Gender)

.End ()

.Separator ()

.Input ()

.Field (x => x.DateOfBirth).Width (90)

.End ()

.Input ()

.Field (x => x.Pictures)

.End ();