Algorithmes :

```
CreerLigne(x1 : double, y1 : double, x2 : double, y2 : double, couleur : chaîne,
epaisseurTrait : double) : Line
{
	retourner new Line
	{
		X1 ← x1
		Y1 ← y1
		X2 ← x2
		Y2 ← y2
		Stroke ← new
		SolidColorBrush((Color)ColorConverter.ConvertFromString(couleur))
		StrokeThickness ← epaisseurTrait
	}
}

CreerPointArret(longueur : double, hauteur : double, couleur : chaine,
epaisseurTrait : double, etiquette : chaine) : Ellipse
{
	retourner new Ellipse
	{
		Width ← longueur
		Height ← hauteur
		Fill ← new
		SolidColorBrush((Color)ColorConverter.ConvertFromString(couleur))
		Stroke ← new
		SolidColorBrush((Color)ColorConverter.ConvertFromString(couleur))
		StrokeThickness ← epaisseurTrait
		Tag ← etiquette
	}
}

CreerEtiquetteNomArret(nom : chaine, taillePolice : double, estPoleEchange :
booléen) : Label
{
	Si estPoleEchange est Vrai Alors
	{
		retourner new Label
		{
			Content ← nom
			FontSize ← taillePolice
			HorizontalContentAlignment ← HorizontalAlignment.Center,
			VerticalContentAlignment ← VerticalAlignment.Center
			FontWeight ← FontWeights.Bold
		}

	}
	Sinon
	{
		retourner new Label
		{
			Content ← nom
			FontSize ← taillePolice
			HorizontalContentAlignment ← HorizontalAlignment.Center,
			VerticalContentAlignment ← VerticalAlignment.Center
		}
	}
	FinSi
}
```

```
CreerEtiquetteCorrespondance(taillePolice : double, estPoleEchange : booléen) :
Label
{
	Si estPoleEchange est Vrai Alors
	{
		retourner new Label
		{
			Content ← "C"
			FontSize ← taillePolice
			FontWeight ← FontWeights.Bold
			Foreground ← Brushes.Black
		}
	}
	Sinon
	{
		retourner new Label
		{
			Content ← "C"
			FontSize ← taillePolice
			FontWeight ← FontWeights.Bold
			Foreground ← Brushes.White
		}
	}
	FinSi
}

CreerRectangleNoLigne(longueur : double, hauteur : double, couleur : chaine,
bordureX : double, bordureY : double, etiquette : chaine) : Rectangle
{
	retourner new Rectangle
	{
		Width ← longueur
		Height ← hauteur
		Stroke ← new
		SolidColorBrush((Color)ColorConverter.ConvertFromString(couleur))
		Fill ← new
		SolidColorBrush((Color)ColorConverter.ConvertFromString(couleur))
		RadiusX ← bordureX
		RadiusY ← bordureY
		Tag ← etiquette
		HorizontalAlignment ← HorizontalAlignment.Center
		VerticalAlignment ← VerticalAlignment.Center
	}
}

CreerEtiquetteNoLigne(longueur : double, hauteur : double, contenu : chaine,
taillePolice : double, etiquette : chaine) : Label
{
	retourner new Label
	{
		Width ← longueur
		Height ← hauteur
		Content ← contenu
		FontSize ← taillePolice
		FontWeight ←FontWeights.Bold
		Foreground ←Brushes.White
		HorizontalAlignment ← HorizontalAlignment.Center
		VerticalAlignment ← VerticalAlignment.Center
		Tag ← etiquette
```

```
	}
}

ModeliserPoleEchange(nomArret : chaine, tpNomArret : chaine,
tpEtiquetteCorrespondance : chaine, estPoleEchange : booléen, couleurPoleEchange :
chaine, longueurPointPoleEchange : double, hauteurPointPoleEchange : double,
epaisseurTraitPoleEchange : double)
{
	Si estPoleEchange est Vrai Alors
	{
		CreerPointArret(longueurPointPoleEchange, hauteurPointPoleEchange,
		couleurPoleEchange, epaisseurTraitPoleEchange, nomArret)
		CreerEtiquetteNomArret(nomArret, tpNomArret, estPoleEchange)
		CreerEtiquetteCorrespondance(tpEtiquetteCorrespondance,
		estPoleEchange)

	}
}
```