

# Package “fast-diagram.sty”

Raphaël ALLAIS

27 février 2013

# Table des matières

Package "fast-diagram.sty"	Introduction (p.3)	Le pourquoi du comment (p.3)
		Petit rappel (p.3)
		Limitations - Perspectives (p.3)
	Un exemple presque complet (p.4)	
	Mise en place du package (p.6)	Installation (p.6)
		Packages requis (p.6)
		Appel du package "fast-diagram.sty" (p.6)
	Détail des commandes (p.7)	Environnement "fast" (p.7)
		Principe des commandes (p.7)
		Fonction technique (p.8)
		Solution technique (p.9)
		Fonction vide (p.10)
		Trait continu (p.11)
	Mise en forme (p.12)	Reset (p.12)
		Les dimensions (p.12)
		Couleurs (p.17)
	Jouons avec TikZ (p.20)	TikZ dans le diagramme FAST (p.20)
		Gestion des boîtes (p.21)
		Créer sa propre boîte (p.21)

# 1 Introduction

## 1.1 Le pourquoi du comment

En tant qu'enseignant en sciences industrielles pour l'ingénieur, j'ai réalisé ce package en vue de m'aider à rédiger mes cours. J'ai toutefois essayé de le rendre le plus paramétrable possible afin qu'il puisse être utilisé dans de nombreux cas. (d'autres options/paramètres peuvent éventuellement être rajoutés selon la demande...).

Il s'agit de mon premier package L<sup>A</sup>T<sub>E</sub>X. De plus, ce package fonctionne sur la bibliothèque *TikZ*, que je connaissais jusqu'alors assez mal. Il n'est donc pas exclu qu'il y ait des bugs dans sa conception. Si vous voyez quelque chose d'anormal ou d'incohérent, ou si vous avez des remarques, n'hésitez pas à m'en faire part à l'adresse suivante : [allais.rafael@free.fr](mailto:allais.rafael@free.fr)

Pour la petite histoire, la difficulté pour réaliser ce package a été le caractère récursif du diagramme FAST. En effet, il semblerait que *TikZ* gère très mal la portée locale des variables : Les variables d'une fonction *enfant* écrasaient les variables de sa fonction *parent*. Cela posait des problèmes sur l'alignement des boîtes. D'autre part, *TikZ* propose déjà des diagrammes en arborescence, mais je n'ai pas su créer mes propres fonctions par dessus.

Merci à Yannick Le Bras, Robert Papanicola et Xavier Pessoles pour leur aide et leurs conseils.

## 1.2 Petit rappel

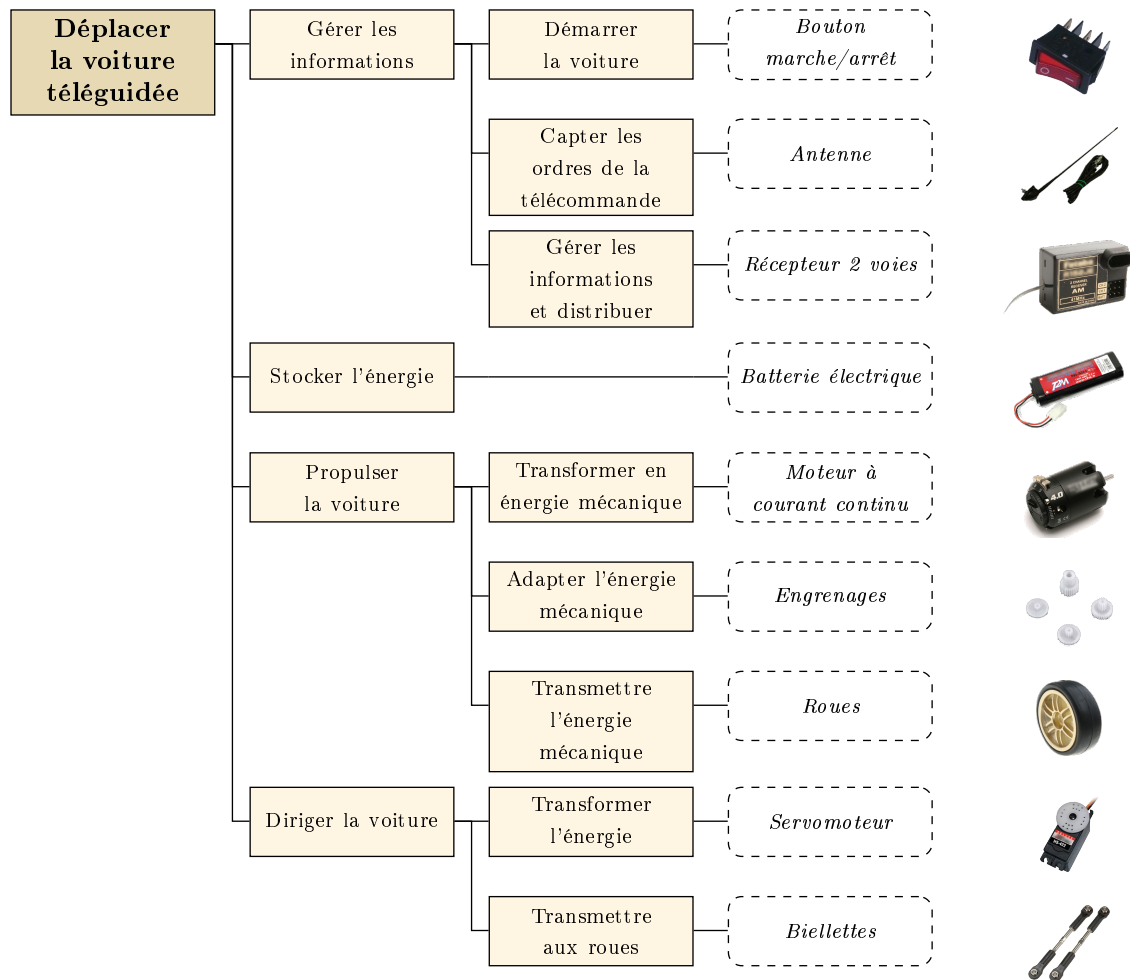
Le diagramme “*Function Analysis System Technique*”, plus couramment appelé “*diagramme FAST*” est un outil de **l'analyse fonctionnelle**, permettant de décrire et de décomposer hiérarchique une *fonction de service* en sous-fonctions, appelées *fonctions techniques*. L'aboutissement d'un tel schéma doit être un ensemble de choix concrets appelés “*solutions techniques*”. Historiquement, ce type de diagramme a été un passage indispensable dans le domaine de la conception et la rédaction des cahiers des charges. Aujourd'hui, une approche plus globale (mais partiellement basée sur des concepts similaires) est proposée au travers des diagrammes [SysML](#).

Pour plus de détail, n'hésitez pas à consulter les nombreux cours qui existent sur Internet.

## 1.3 Limitations - Perspectives

Le package a été écrit pour répondre **aux principales attentes** du diagramme FAST. Il n'est cependant pas complet. Il n'est, par exemple, pas possible de relier **automatiquement** une solution technique commune à plusieurs fonctions techniques. Cette possibilité n'est toutefois pas exclue puisque les commandes de *TikZ* sont autorisées à l'intérieur de l'environnement (voir 6) et rien n'empêche de le faire “à la main”. N'hésitez donc pas à me faire part d'éventuelles autres fonctions à mettre en place.

## 2 Un exemple presque complet



L'exemple ci-dessus est donné par le code suivant :

```

\begin{center}
\footnotesize

\definecolor{fastCouleurFondFS}{rgb}{0.90,0.85,0.70}
\definecolor{fastCouleurFondFT}{rgb}{1,0.96,0.89}
\definecolor{fastCouleurFondST}{rgb}{1,1,1}
\renewcommand*{\fastHauteurBoite}{2.6em}
\renewcommand*{\fastDecalageTrait}{-1.3em}
\renewcommand*{\fastEspaceColonne}{9em}

\begin{fast}{Déplacer la voiture téléguidée}
  \FT{Gérer les informations}
    {\FT{Démarrer la voiture}
      {\ST{Bouton marche/arrêt}
        [\FV{\includegraphics[height=1cm]{./sources_help/images/bouton.png}}]}
    }
  \FT{Capter les ordres de la télécommande}
    {\ST{Antenne}
      [\FV{\includegraphics[height=1cm]

```

```

        {./sources_help/images/antenne.png}}]
    }
\FT{Gérer les informations et distribuer}
    {\ST{Récepteur 2 voies}
        [\FV{\includegraphics[height=1cm]
            {./sources_help/images/recepteur.png}}]}
    }
\FT{Stocker énergie}
    {\trait{
        \ST{Batterie électrique}
            [\FV{\includegraphics[height=1cm]
                {./sources_help/images/batterie.png}}]}
    }
\FT{Propulser la voiture}
    {\FT{Transformer en énergie mécanique}
        {\ST{Moteur à courant continu}
            [\FV{\includegraphics[height=1cm]
                {./sources_help/images/moteur.png}}]}
        }
    \FT{Adapter l'énergie mécanique}
        {\ST{Engrenages}
            [\FV{\includegraphics[height=1cm]
                {./sources_help/images/pignons.png}}]}
        }
    \FT{Transmettre l'énergie mécanique}
        {\ST{Roues}
            [\FV{\includegraphics[height=1cm]
                {./sources_help/images/roue.png}}]}
        }
\FT{Diriger la voiture}
    {\FT{Transformer l'énergie}
        {\ST{Servomoteur}
            [\FV{\includegraphics[height=1cm]
                {./sources_help/images/servomoteur.png}}]}
        }
    \FT{Transmettre aux roues}
        {\ST{Biellettes}
            [\FV{\includegraphics[height=1cm]
                {./sources_help/images/biellettes.png}}]}
        }
}
}
\end{fast}
\fastReset
\end{center}

```

## 3 Mise en place du package

### 3.1 Installation

Le package s’installe comme n’importe quel autre. Après l’avoir téléchargé, copier le :

- soit dans le dossier du document que vous êtes en train de rédiger (c’est une méthode facile, mais il ne sera valable que pour ce document-là)
- soit dans un des dossiers par défaut de latex. L’emplacement de ces dossiers dépendent du logiciel et du système d’exploitation utilisé (Windows, Mac, Linux, etc.).


### 3.2 Packages requis

Pour que le package fonctionne, vous devez déjà avoir les packages suivants d’installés :

- **TikZ** : Package de dessin vectoriel sur lequel repose le diagramme fast,
- **ifthen** : Package permettant une compilation à choix multiple,
- **relsize** : Package permettant de gérer les longueurs relatives (em, ...)
- **xarg** : Package permettant de créer des commandes à plusieurs arguments optionnels.

### 3.3 Appel du package “fast-diagram.sty”

L’appel du package se fait simplement en écrivant dans l’entête du document :

 Code  
| `\usepackage{fast-diagram}`

Afin d’éviter d’éventuels conflits entre packages, toutes les commandes utilisées ici sont précédées du préfixe **fast** (par exemple `\fastFT` pour désigner la fonction technique FT). Pour la mise en place de raccourcis, l’option `[raccourcis]` peut être apportée dans le package de la manière suivante :

 Code  
| `\usepackage[raccourcis]{fast-diagram}`

Les raccourcis seront développés plus tard.

## 4 Détail des commandes

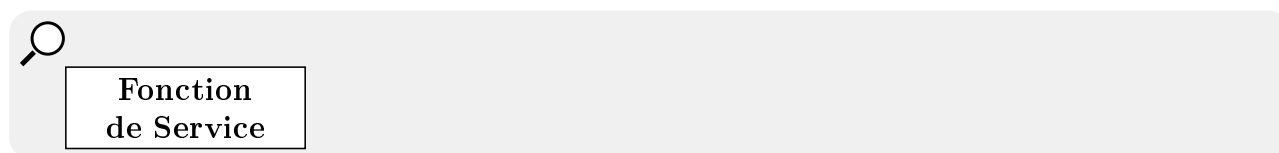
### 4.1 Environnement “fast”

Le diagramme fast est placé dans l’environnement `\begin{fast}... \end{fast}`. Cet environnement prend comme argument la *fonction de service* que l’on souhaite développer.

#### Code

```
\begin{fast}{Fonction de Service}
    %Votre diagramme FAST
\end{fast}
```

Ce qui donne :



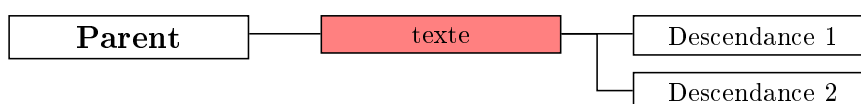
A l’intérieur de l’environnement `fast`, on va alors venir placer chacune des fonctions techniques, solutions techniques, etc. Ces commandes vont être décrites dans les paragraphes suivants.

### 4.2 Principe des commandes

Une fois l’environnement `fast` ouvert, le but du jeu va être de créer des fonctions (c’est à dire des “*boites*”) à l’intérieur, reliées entre elles de manière hiérarchique.

Il existe plusieurs “boites” différentes qui seront chacune développées dans les paragraphes suivants.

Chaque boite possède un “**parent**” en amont, un “**texte**” à l’intérieur et éventuellement une ou plusieurs “**descendances**” en aval.



Le texte de chaque fonction est passé en premier argument de la commande.

On parlera de fonctions “*sœurs*” lorsque ces fonctions sont en parallèles, issues d’un même parent. Les commandes permettant de créer plusieurs fonctions sœurs sont placées les unes à la suite des autres.

#### Code

```
\begin{fast}{PARENT}
    \une_fonction{texte}{Descendance de la fonction}
    \une_fonction_soeur{texte}{Descendance de la fonction soeur}
\end{fast}
```

On parlera de fonctions “*filles*” les fonctions descendant d’un parent. Les fonctions filles sont passées en deuxième argument de leur parent.

## Code

```
\begin{fast}{PARENT}
  \une_fonction{texte}{
    \une_fonction_fille{texte}{descendance}
    \une_autre_fonction_fille{texte}{descendance}
  }
\end{fast}
```

En pratique, la descendance peut être n'importe quelle fonction *TikZ* (voir 6). Elle peut également ne rien comporter.

La suite de ce chapitre va présenter les différentes fonctions disponibles.

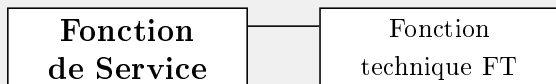
### 4.3 Fonction technique

`\fastFT` (raccourci : `\FT`) est une commande “de base” du diagramme FAST. Elle s’emploie de la manière suivante :

## Code

```
\begin{fast}{Fonction de Service}
  \fastFT{Fonction technique FT}
  {
    %Descendance
  }
\end{fast}
```

Ce qui donne :



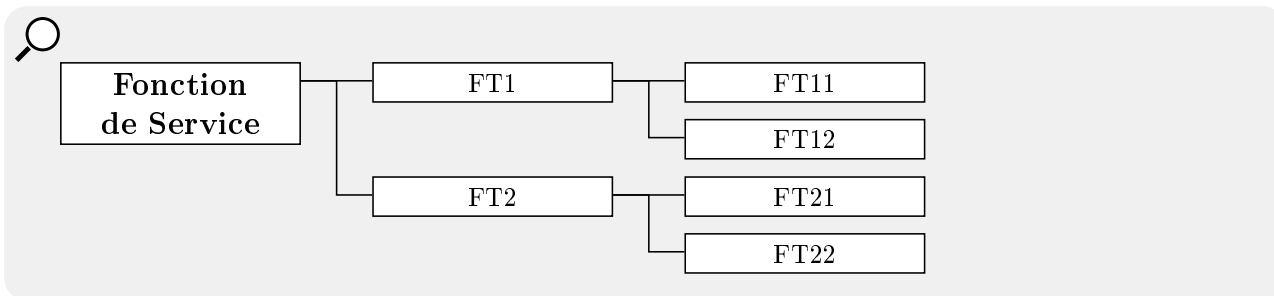
Voici un exemple d'utilisation en série et en parallèle :

## Code

```
\begin{fast}{Fonction de Service}
  \fastFT{FT1}
  {
    \fastFT{FT11}{ }
    \fastFT{FT12}{ }
  }
  \fastFT{FT2}
  {
    \fastFT{FT21}{ }
    \fastFT{FT22}{ }
  }
\end{fast}
```


Ce qui donne :





Si le premier argument est vide, cela revient à faire un trait horizontal, au même titre que que la fonction `\fastFTrait` (voir 4.6).

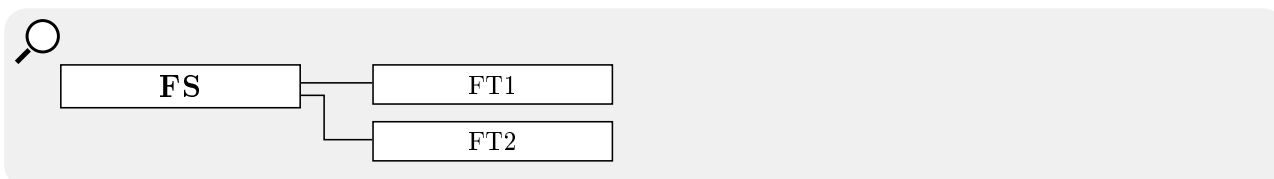
La commande `\fastFT` peut également prendre un mot-clé en options : le mot clé `[ou]` ; il décale légèrement le connecteur pour représenter un liaison “ou” (voir la mise en forme au paragraphe 5.2).

 **Code**

```


\begin{fast}{FS}
  \FT{FT1}
  {
    \fastFT{FT1}{}
    \fastFT[ou]{FT2}{}
  }
\end{fast}
  
```

Ce qui donne :



## 4.4 Solution technique

`\fastST` (raccourci : `\ST`) prend un seul argument : le contenu de la solution technique.

 **Code**

```

\begin{fast}{Fonction de Service}
  \fastST{Solution technique}
\end{fast}
  
```

Ce qui donne :



Normalement, la solution technique correspond à la fin d’une branche du diagramme FAST. C’est pourquoi elle ne requière pas d’autre argument. Toutefois, pour des besoins spécifiques (commentaire, image, etc.), on peut lui rajouter une descendance en option :

## Code

```
\begin{fast}{Fonction de Service}
  \fastST{Solution technique}[\fastVide{Commentaire...}]
\end{fast}
```

Ce qui donne :



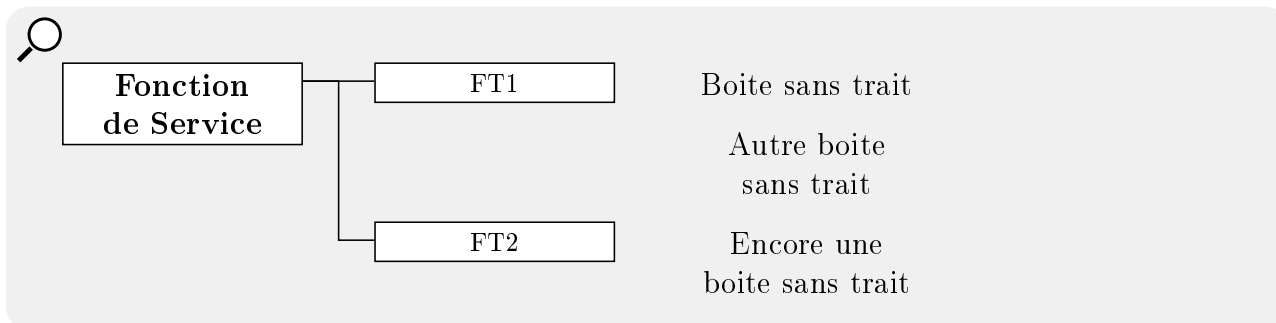
## 4.5 Fonction vide

`\fastVide` (raccourci : `\FV`) permet de faire une case sans connecteur ni bordure.

## Code

```
\begin{fast}{Fonction de Service}
  \fastFT{FT1} {
    \fastVide{Boite sans trait}
    \fastVide{Autre boite sans trait}
  }
  \fastFT{FT2}{ \fastVide{Encore une boite sans trait}}
\end{fast}
```

Ce qui donne :

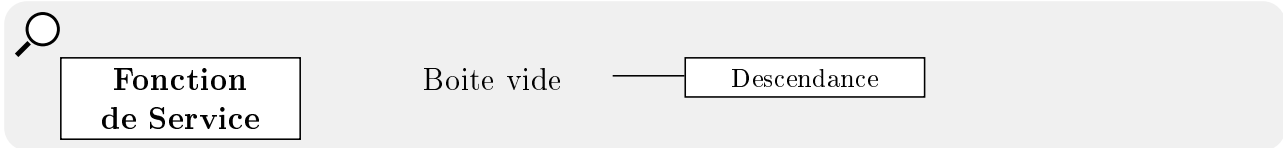


Tout comme pour la boîte “solution technique”, cette fonction est destinée à être en bout de branche du diagramme. On ne demande donc pas de descendance. Toutefois, on peut la lui proposer en argument optionnel :

## Code

```
\begin{fast}{Fonction de Service}
  \fastVide{Boite vide}[\fastFT{Descendance}{}]
\end{fast}
```

Ce qui donne :



## 4.6 Trait continu

`\fastTrait` (raccourci : `\trait`) représente un simple trait. Il permet en effet de tracer un connecteur directement de la colonne  $(n - 1)$  à  $(n + 1)$ , en “sautant” la colonne  $(n)$ . Le seul argument demandé est la descendance de ce connecteur. La fonction technique `\fastFT` avec un premier argument vide réalise la même chose.

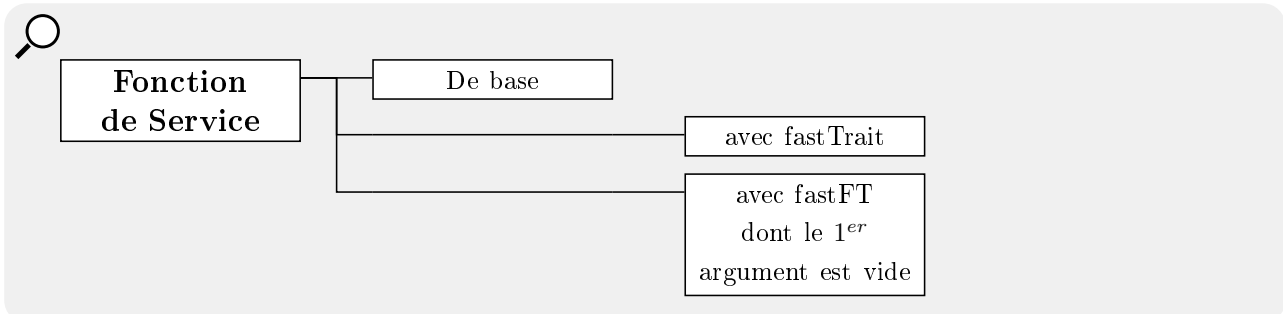
### Code

```

\begin{fast}{Fonction de Service}
  \fastFT{De base}{}
  \fastTrait {
    \fastFT{avec fastTrait}{}
  }
  \fastFT{} {
    \fastFT{avec fastFT dont le $1^{er}$ argument est vide}{}
  }
\end{fast}

```

Ce qui donne :



## 5 Mise en forme

### 5.1 Reset

`\fastReset` permet de remettre les paramètres par défaut.

### 5.2 Les dimensions

Les dimensions du diagramme sont réglées via plusieurs commandes. En voici la liste :

- `\fastInterligne` : espace entre le bas de la boîte la plus grande de la ligne en cours, et le haut des boîtes de la ligne suivante. Ce nombre doit être positif. (Par défaut : 0.5 em)
- `\fastLargeurBoite` : largeur des boîtes (Par défaut : 7 em)
- `\fastHauteurBoite` : hauteur **minimum** des boîtes (Par défaut : 0)
- `\fastEspaceColonne` : distance entre le coin supérieur gauche d’une boîte et le coin supérieur gauche de sa voisine. (Par défaut : 10 em)
- `\fastDecalageTrait` : permet de décaler le connecteur par rapport au haut de la boîte. (Par défaut : -0.6 em)
- `\fastEpaisseurTraits` : épaisseur des traits (bordures et connecteurs). (Par défaut : 0.05 em)
- `\fastDecalageOuVertical` : Décalage vertical du connecteur “OU”. (Par défaut : 0.4 em)
- `\fastDecalageOuHorizontal` : Décalage horizontal du connecteur “OU”. (Par défaut : -0.4 em)

Les deux dernières fonctions peuvent être utiles si plusieurs connecteur “OU” sont utilisés sur la même ligne.

Toutes ces commandes peuvent être redéfinies via la fonction la fonction `\renewcommand` (ou `\renewcommand*`). Voici ci-dessous une série d’exemples illustrant chacune de ces fonctions.

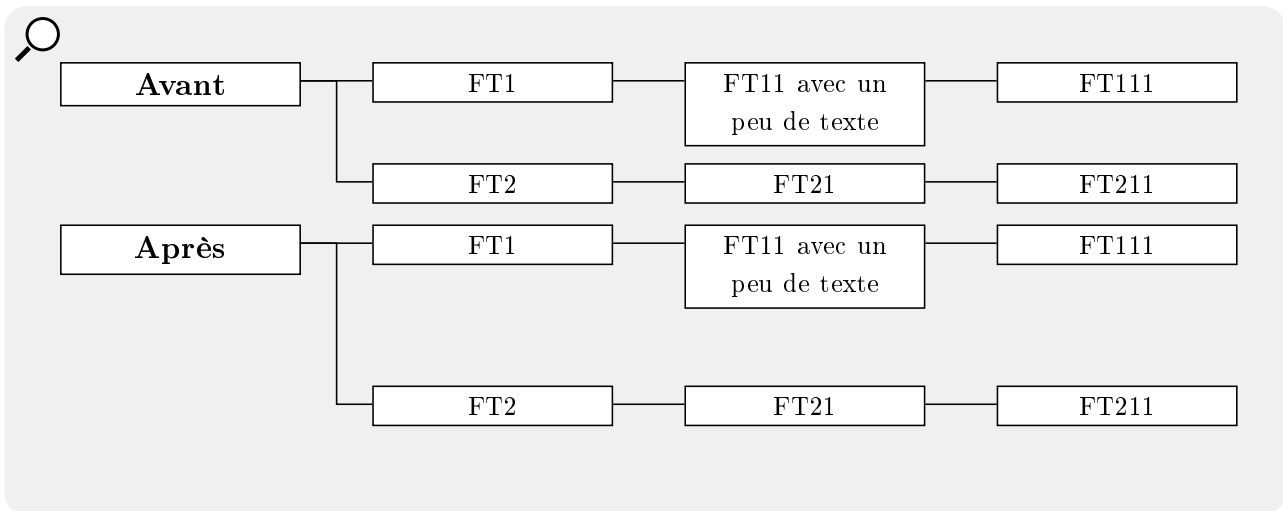
#### 5.2.1 Exemple : interlignes

##### Code

```
\begin{fast}{Avant} %Interligne par défaut
  \fastFT{FT1}{
    \fastFT{FT11 avec un peu de texte}{
      \fastFT{FT111}{}}}
  \fastFT{FT2}{
    \fastFT{FT21}{
      \fastFT{FT211}{}}}
\end{fast}

\renewcommand*{\fastInterligne}{1cm} %Nouvel interligne
\begin{fast}{Après}
  \fastFT{FT1}{
    \fastFT{FT11 avec un peu de texte}{
      \fastFT{FT111}{}}}
  \fastFT{FT2}{
    \fastFT{FT21}{
      \fastFT{FT211}{}}}
\end{fast}
\fastReset %Remise à zéro
```

Ce qui donne :



### 5.2.2 Exemple : largeur des boîtes

#### Code

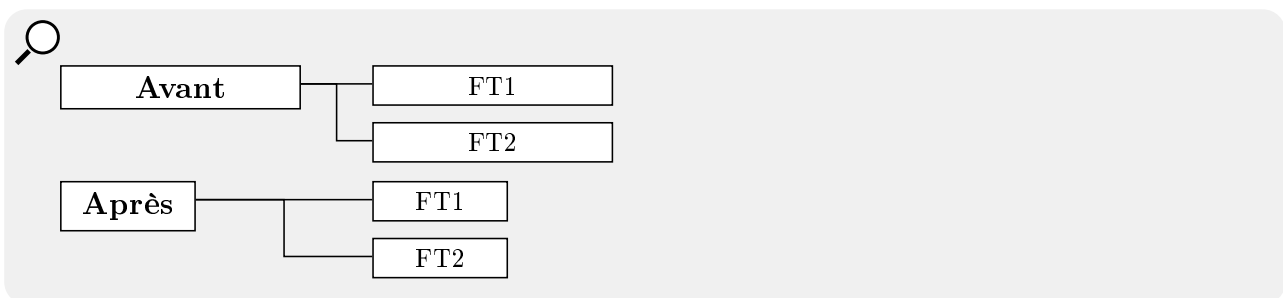
```

\begin{fast}{Avant}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
\end{fast}

\renewcommand*{\fastLargeurBoite}{1.5cm} %Nouvelle largeur de boite
\begin{fast}{Après}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
\end{fast}
\fastReset

```

Ce qui donne :

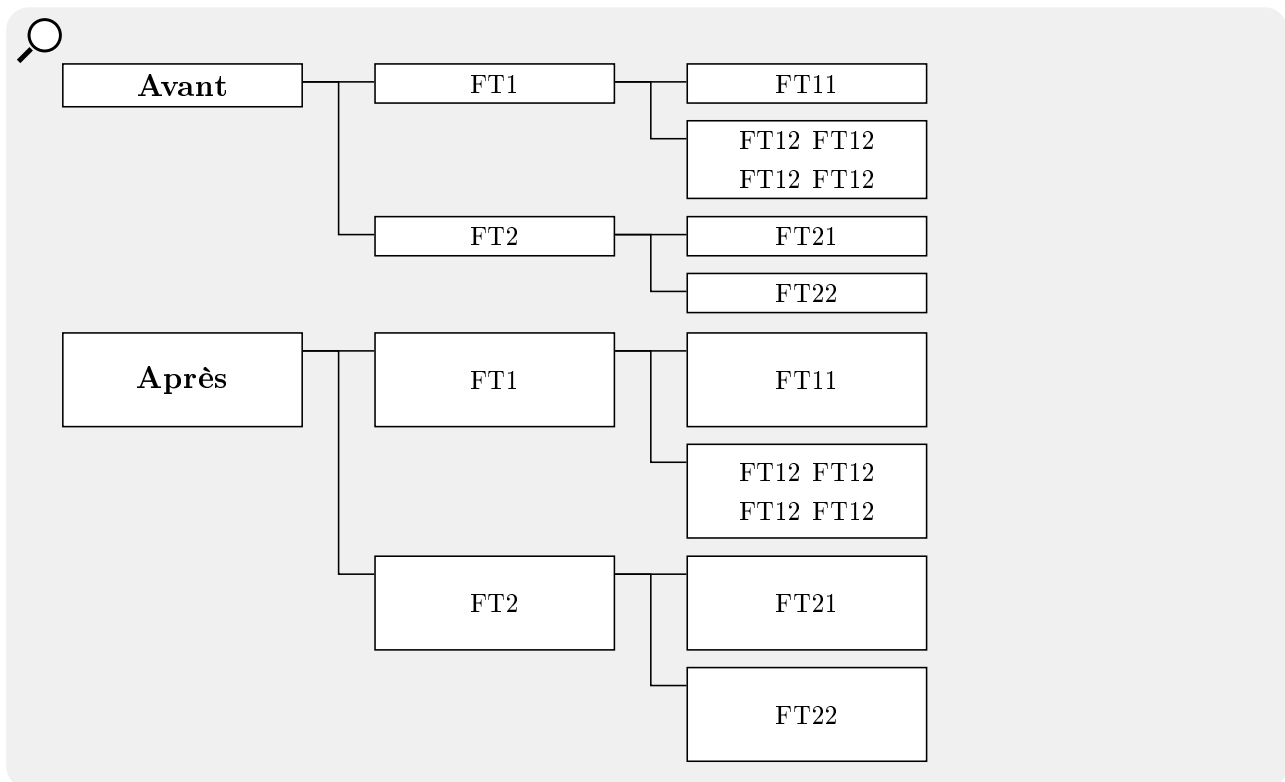


### 5.2.3 Exemple : hauteur des boîtes

#### Code

```
\begin{fast}{Avant}
  \fastFT{FT1}{ \FT{FT11}{ }
                \FT{FT12 FT12 FT12 FT12}{ }}
  \fastFT{FT2}{ \FT{FT21}{ }
                \FT{FT22}{ }}
\end{fast}
\renewcommand*{\fastHauteurBoite}{3em}
\begin{fast}{Après}
  \fastFT{FT1}{ \FT{FT11}{ }
                \FT{FT12 FT12 FT12 FT12}{ }}
  \fastFT{FT2}{ \FT{FT21}{ }
                \FT{FT22}{ }}
\end{fast}
\fastReset
```

Ce qui donne :



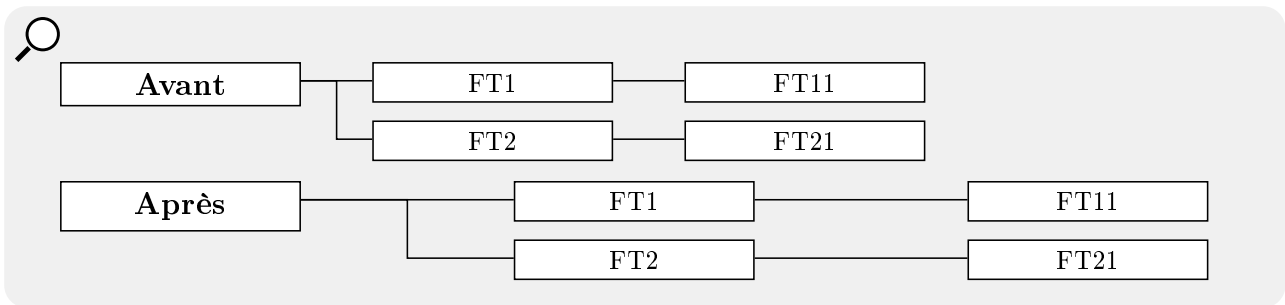
## 5.2.4 Exemple : espace entre colonnes

### Code

```
\begin{fast}{Avant}
  \fastFT{FT1}{
    \fastFT{FT11}{}}
  \fastFT{FT2}{
    \fastFT{FT21}{}}
\end{fast}

\renewcommand*{\fastEspaceColonne}{6cm} %Nouvel espace inter-colonnes
\begin{fast}{Après}
  \fastFT{FT1}{
    \fastFT{FT11}{}}
  \fastFT{FT2}{
    \fastFT{FT21}{}}
\end{fast}
\fastReset
```

Ce qui donne :



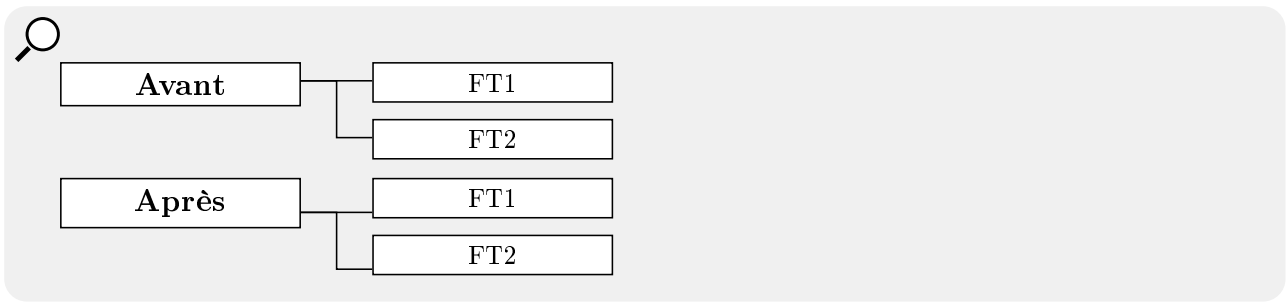
## 5.2.5 Exemple : décalage des connecteurs

### Code

```
\begin{fast}{Avant}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
\end{fast}

\renewcommand*{\fastDecalageTrait}{-13pt} %Nouveau décalage des connecteur
\begin{fast}{Après}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
\end{fast}
\fastReset
```

Ce qui donne :



### 5.2.6 Exemple : épaisseur des traits

#### Code

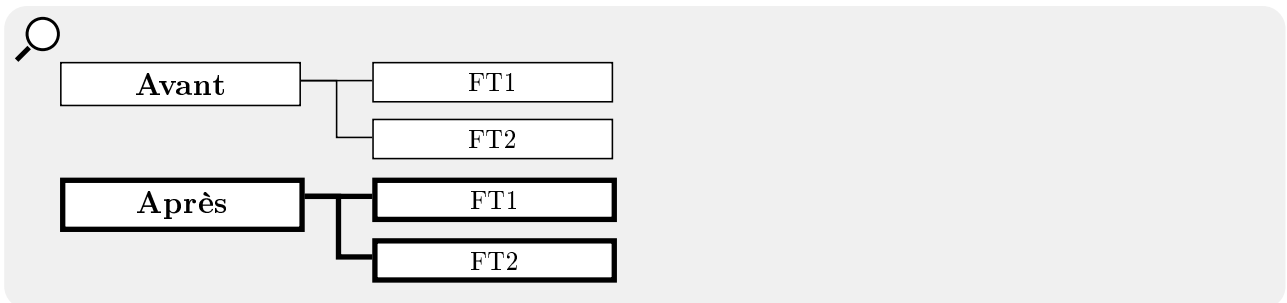
```

\begin{fast}{Avant}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
\end{fast}

\renewcommand*{\fastEpaisseurTraits}{2pt} %Nouvelle épaisseur de traits
\begin{fast}{Après}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
\end{fast}
\fastReset

```

Ce qui donne :





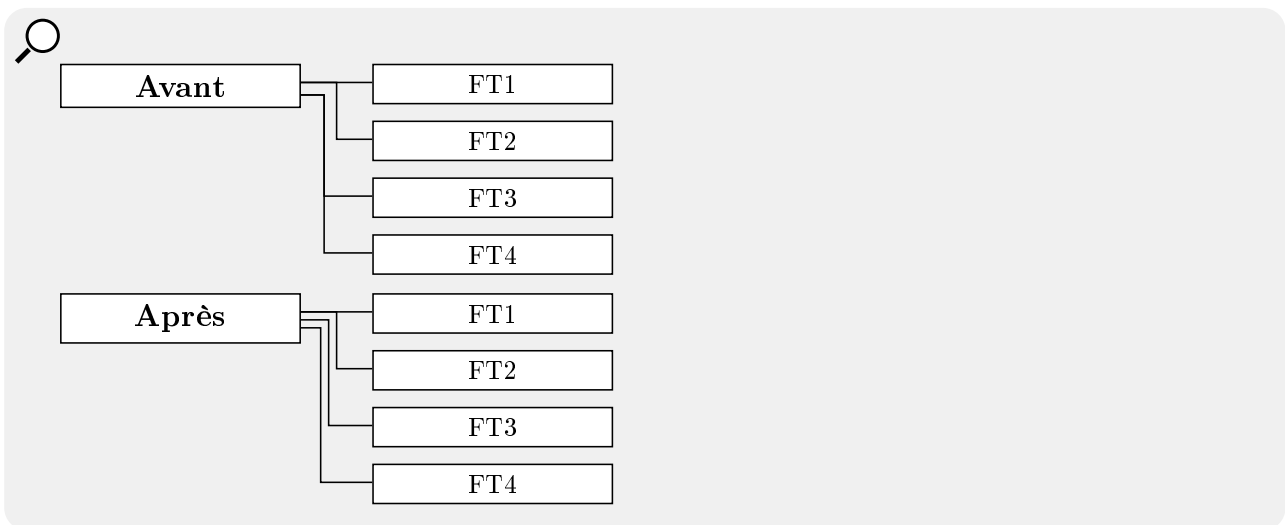
## 5.2.7 Exemple : Décalage des connecteur “OU”

### Code

```
\begin{fast}{Avant}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
  \fastFT[ou]{FT3}{}
  \fastFT[ou]{FT4}{}
\end{fast}

\renewcommand*{\fastDecalageOuVertical}{3pt} %Redécalage vertical...
\renewcommand*{\fastDecalageOuHorizontal}{-3pt} %... et horizontal du 1er "OU"
\begin{fast}{Après}
  \fastFT{FT1}{}
  \fastFT{FT2}{}
  \fastFT[ou]{FT3}{}
  \renewcommand{\fastDecalageOuVertical}{6pt} %Décalage vertical...
  \renewcommand{\fastDecalageOuHorizontal}{-6pt} %...et horizontal...
  \fastFT[ou]{FT4}{} % ...du 2eme "OU"
\end{fast}
\fastReset
```

Ce qui donne :



## 5.3 Couleurs

Il est possible de modifier les couleurs de plusieurs éléments tels que :

- la **fonction de service** (la première case),
- les **fonctions techniques**,
- les **solutions techniques**,
- les **boîtes vides**,
- les **connecteurs**.

Pour chacun des quatre premiers points précédents, on peut définir :

- la couleur du **texte**,
- la couleur du **fond** (sauf boîte vide),
- la couleur du **cadre** (sauf boîte vide).

Tout cela donne un total de 11 couleurs, définies par les noms suivants :

- `fastCouleurTexteFS` : Couleur du texte de la fonction de service (la 1<sup>ere</sup> boite),
- `fastCouleurBorduresFS` : Couleur de bordure de la fonction de service (la 1<sup>ere</sup> boite),
- `fastCouleurFondFS` : Couleur du fond de la fonction de service (la 1<sup>ere</sup> boite),
- `fastCouleurTexteFT` : Couleur du texte des fonctions techniques,
- `fastCouleurBorduresFT` : Couleur de bordure des fonctions techniques,
- `fastCouleurFondFT` : Couleur du fond des fonctions techniques,
- `fastCouleurTexteST` : Couleur du texte des solutions techniques,
- `fastCouleurBorduresST` : Couleur de bordure des solutions techniques,
- `fastCouleurFondST` : Couleur du fond des solutions techniques,
- `fastCouleurTexteFV` : Couleur du texte de la fonction de boite vide,
- `fastCouleurConnecteurs` : Couleur des connecteurs.

Toutes ces couleurs peuvent être redéfinies par la fonction `\definecolor` (voir le package `xcolor`) :

### Code

```

\definecolor{fastCouleurTexteFS} {named} {white}
\definecolor{fastCouleurBorduresFS} {named} {red}
\definecolor{fastCouleurFondFS} {named} {red}

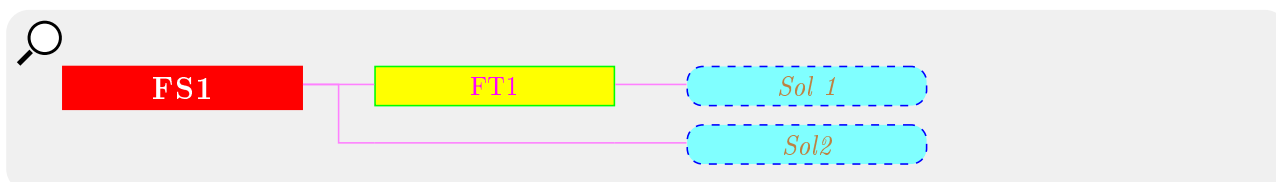
\definecolor{fastCouleurTexteFT} {rgb} {1,0,1}
\definecolor{fastCouleurBorduresFT} {rgb} {0,1,0}
\definecolor{fastCouleurFondFT} {rgb} {1,1,0}

\definecolor{fastCouleurTexteST} {named} {brown}
\definecolor{fastCouleurBorduresST} {named} {blue}
\definecolor{fastCouleurFondST} {rgb} {0.5,1,1}

\definecolor{fastCouleurConnecteurs} {rgb} {1,0.5,1}
\begin{fast}{FS1}
  \fastFT{FT1}{
    \fastST{Sol 1}}
  \fastFT{}{
    \fastST{Sol2}}
\end{fast}
\fastReset

```

Ce qui donne :



Pour aller plus vite, trois commandes servent de raccourci :

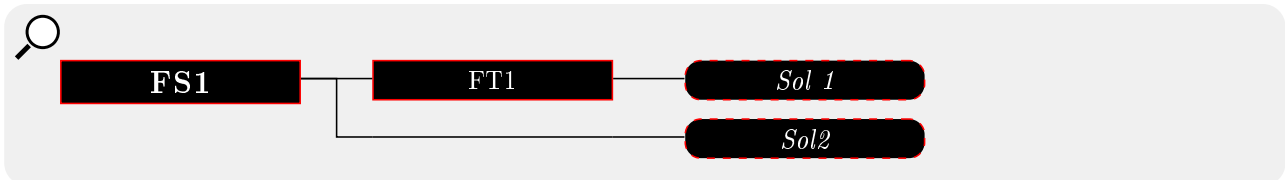
- `\fastSetCouleurBordures [type] {couleur}` : permet de changer la couleur de toutes les bordures,
- `\fastSetCouleurTexte [type] {couleur}` : permet de changer la couleur de tout le texte,
- `\fastSetCouleurTraits [type] {couleur}` : permet de changer la couleur de toutes les lignes (bordures + connecteurs),
- `\fastSetCouleurFond [type] {couleur}` : permet de changer la couleur du fond de toutes les boites,

où `[type]` est le type d'affectation (*rgb,cmymk,named*(par défaut),...) et `[couleur]` est la couleur, relativement à `[type]` (voir `\definecolor` du package `xcolor`).

## Code

```
\fastSetCouleurBordures{red}  
\fastSetCouleurTexte[rgb]{1,1,1}  
\fastSetCouleurFond{black}  
\begin{fast}{FS1}  
  \fastFT{FT1}{  
    \fastST{Sol 1}}  
  \fastFT{}{  
    \fastST{Sol2}}  
\end{fast}  
\fastReset
```

Ce qui donne :



## 6 Jouons avec TikZ

### 6.1 TikZ dans le diagramme FAST

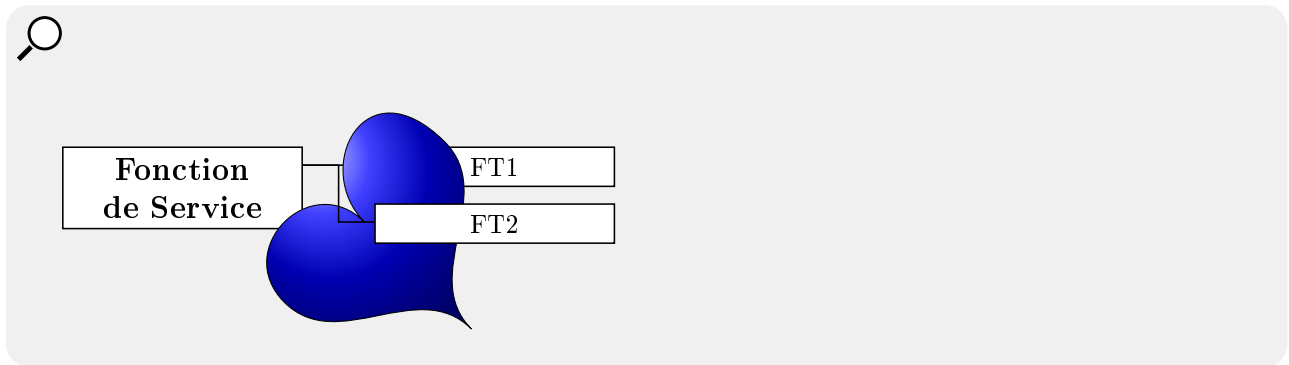
L'environnement FAST est un environnement *TikZ*. Il est donc possible d'y ajouter n'importe quelle fonction de dessin de *TikZ*. Il en est de même pour les descendances des fonctions.

#### Code

```
\begin{fast}{Fonction de Service}
  \FT{FT1}{\draw [shift={(4,-1)},rotate=45,scale=0.5,ball color=blue]
    (0,0) .. controls +(0,2) and +(0,3) .. (3,0)
    .. controls +(0,-2) and +(0,2) .. (0,-4)
    .. controls +(0,2) and +(0,-2) .. (-3,0)
    .. controls +(0,2) and +(0,2) .. (0,0);
  }
  \FT{FT2}{}
\end{fast}
```

*%Exemple pris dans "TikZ pour l'impatient"*

Ce qui donne :



Il est à noter que dans l'exemple précédent, la seconde ligne du diagramme ne tient pas compte de la "place" que prend notre dessin. Pour que ce soit le cas, il faut que la descendance (c'est à dire le dessin) "marque" sa place en créant une coordonnée correspondant au point le plus bas du dessin. C'est sur ce point que la seconde ligne va se baser.

Ce point doit être enregistré dans la variable `BoiteMinimums` de la manière suivante :

#### Code

```
\coordinate (BoiteMinimums) at (X,Y);
```

où le couple  $(X, Y)$  est la coordonnées du minimum.

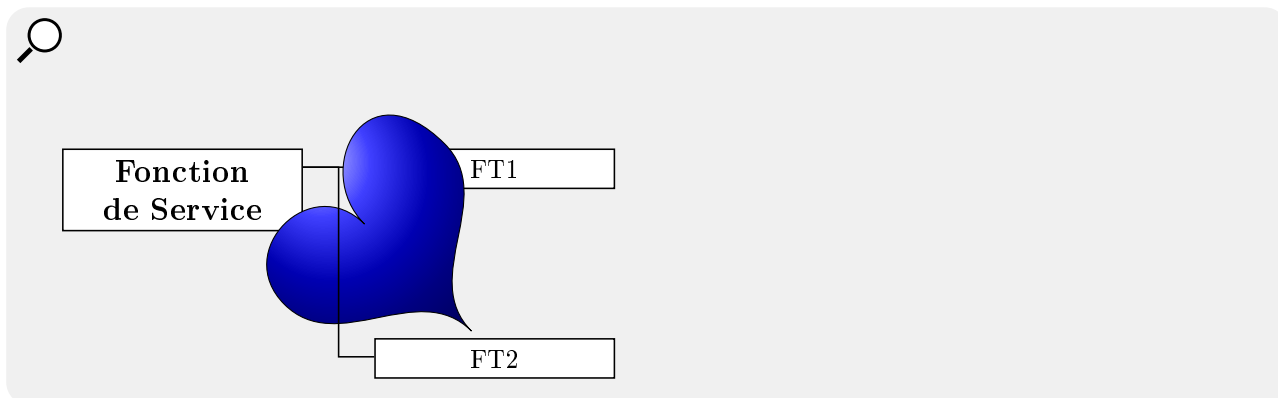
Par exemple :

#### Code

```
\begin{fast}{Fonction de Service}
  \FT{FT1}{\draw [shift={(4,-1)},rotate=45,scale=0.5,ball color=blue]
    (0,0) .. controls +(0,2) and +(0,3) .. (3,0)
    .. controls +(0,-2) and +(0,2) .. (0,-4)
    .. controls +(0,2) and +(0,-2) .. (-3,0)
    .. controls +(0,2) and +(0,2) .. (0,0);
    \coordinate (BoiteMinimums) at (0,-2.5);
  }
  \FT{FT2}{}
\end{fast}
```

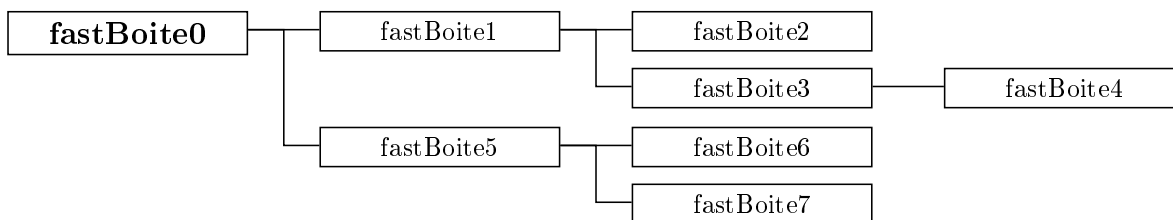
*%Exemple pris dans "TikZ pour l'impatient"*

Ce qui donne :



## 6.2 Gestion des boîtes

Les boîtes créées dans le diagramme FAST sont réalisées par la fonction `\node` de *TikZ*. Ces boîtes sont nommées sous la forme suivante : `\fastBoiteX` où `X` est remplacé par le numéro de la boîte. Ce numéro est défini par ordre de création des boîtes : de gauche à droite, de haut en bas. Voici un exemple faisant apparaître le nom des boîtes :

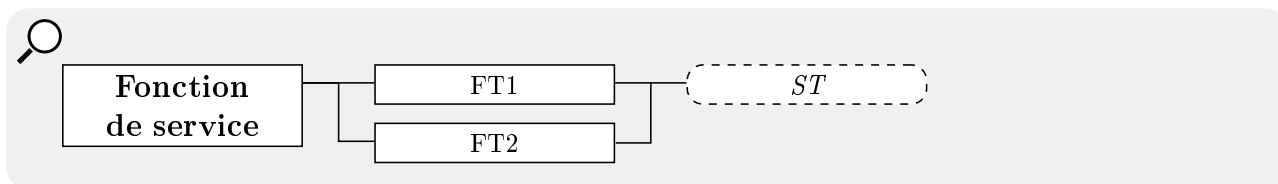


Partant de là, il est alors possible de réaliser des modifications manuelles sur le diagramme. Par exemple, pour avoir une solution technique commune à deux fonctions techniques :

### Code

```
\begin{fast}{Fonction de service}
  \fastFT{FT1}{\fastST{ST}}
  \fastFT{FT2}{}
  \draw[line width=\fastEpaisseurTraits]
    (fastBoite3.east) -| ($0.5*(fastBoite2.north west)
    +0.5*(fastBoite1.north east)+(0,\fastDecalageTrait)$);
\end{fast}
```

Ce qui donne :



## 6.3 Créer sa propre boîte

Les boîtes sont à peu près toutes créées sur le même modèle et il est possible d'en créer d'autres :

## Code

```
\newcommand*{\maBoite}[2]{
  \fastAvanceColonne           %On avance d'une colonne
  \addtocounter{cptBoite}{1}   %On incremente le numero de la boite
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %Créer votre boite ici :
  \node [anchor=north west] (noeud \thecptAbscisse) at
    ($(\posX,0)+(BoiteMinimums)$) {#1};
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  \node[inner sep=0,fit=(noeud \thecptAbscisse.north west)
    (noeud \thecptAbscisse.south east)]
    (fastBoite\thecptBoite) {};%Boite de nommage
  \fastTraceConnecteurs
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %Votre descendance :
  #2
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  \fastEnregistreMinimum       %Enregistre le minimum de la boite
  \fastReculerColonne         %Reculer d'une colonne
}
```

Le nœud créé sous la ligne “Créer votre boite ici” est la boite que vous allez afficher. C’est elle que vous allez pouvoir modifier pour l’adapter à vos besoins. Ce nœud doit obligatoirement porter le nom (noeud \thecptAbscisse). Les autres commandes ne doivent pas être changées.

Voici un exemple :

## Code

```
\newcommand*{\maBoite}[2]{
  \fastAvanceColonne           %On avance d'une colonne
  \addtocounter{cptBoite}{1}   %On incremente le numero de la boite
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %Créer votre boite ici
  \node [anchor=north west,draw,rounded corners=3pt,
    aspect=2.5,text=red](noeud \thecptAbscisse)
    at ($(\posX,0)+(BoiteMinimums)$) {#1};
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  \node[inner sep=0,fit=(noeud \thecptAbscisse.north west)
    (noeud \thecptAbscisse.south east)]
    (fastBoite\thecptBoite) {};
  \fastTraceConnecteurs
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %Votre descendance
  #2
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  \fastEnregistreMinimum       %Enregistre le minimum de la boite
  \fastReculerColonne         %Reculer d'une colonne
}

\begin{fast}{Fonction de Service}
  \maBoite{Ma boite}
  {\fastST{Solution}}
  \FT{Fonction}{\maBoite{Ma boite bis}}{}
  \fastFT{Fonction}{}
\end{fast}
```

Ce qui donne :

