

Réconciliation non-fonctionnelle de services

« L'art de la réussite est celui des réconciliations. »

Jacques Deval

| | |
|---|-----|
| IV.1. INTRODUCTION..... | 79 |
| IV.2. RECONCILIATION DE SERVICES | 79 |
| IV.3. RECONCILIATION NON-FONCTIONNELLE DE SERVICES..... | 80 |
| IV.4. RECONCILIATION NON-FONCTIONNELLE DE SERVICES DE TYPE « 1-1 »..... | 84 |
| IV.4.1. Présentation générale de l’approche « 1-1 » | 84 |
| IV.4.2. Algorithme de réconciliation non-fonctionnelle de type « 1-1 » | 84 |
| IV.4.2.1. Étape 1 : Classification des valeurs des exigences non-fonctionnelles..... | 86 |
| IV.4.2.2. Étape 2 : Élimination des services candidats ayant des valeurs aberrantes | 87 |
| IV.4.2.3. Étape 3 : Normalisation et centrage des services candidats par rapport aux exigences non-fonctionnelles de l’activité métier | 89 |
| IV.4.2.4. Étape 4 : Classement et sélection des services candidats..... | 95 |
| IV.5. RECONCILIATION NON-FONCTIONNELLE DE SERVICES DE TYPE « 1-N » | 98 |
| IV.5.1. Présentation générale de l’approche « 1-N » | 98 |
| IV.5.2. Transformation d’une réconciliation non-fonctionnelle de type « 1-N » en une de type « 1-1 » | 100 |
| IV.5.2.1. Étape 1 : Décomposition en plusieurs sous-compositions de deux services | 100 |
| IV.5.2.2. Étape 2 : Calcul de la valeur de la propriété non-fonctionnelle de l’ensemble de la sous-composition..... | 101 |
| IV.6. CONCLUSION | 106 |

IV.1. Introduction

Dans le chapitre précédent, nous avons présenté nos deux prototypes PetalsBPM-NFR et EasierGov-NFR. Le premier nous permet de modéliser graphiquement les processus BPMN 2.0 et d'annoter ses activités avec des exigences non-fonctionnelles. Le deuxième représente notre registre de gouvernance SOA contenant les services Web, leurs propriétés non-fonctionnelles et les modèles de contrats SLA. Notre objectif principal, est de relier le monde métier avec celui technique et donc d'assigner à chaque activité métier de notre processus collaboratif le ou les bons services Web parmi ceux existant dans notre registre de gouvernance SOA. Cette mise en correspondance est appelée « *la réconciliation de services* » que nous présentons dans la Section IV.2. La réconciliation de services se divise en deux parties : la réconciliation fonctionnelle (cf. [Boissel-Dallier, 2012]) et la réconciliation non-fonctionnelle. Dans ces travaux de thèse, nous abordons ce second type de réconciliation (Section IV.3) (cf. Figure 34).

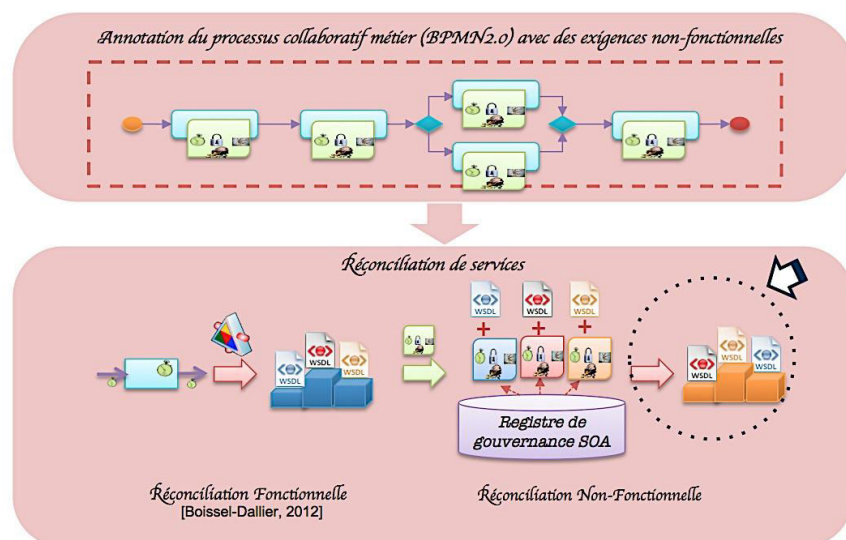


Figure 34 : Réconciliation non-fonctionnelle : sélection et classement des services.

IV.2. Réconciliation de services

L'orchestration du processus métier sous la forme de processus exécutable nécessite un mécanisme de réconciliation entre ceux-ci. Le principe général de ce mécanisme est de trouver, pour chacune des activités du processus métier collaboratif BPMN 2.0, un service ou une composition de services répondant à ses exigences fonctionnelles et non-fonctionnelles. Ces services / compositions de services doivent être classés en fonction de leur proximité avec l'activité métier cible afin que le client puisse sélectionner la solution qui lui semble la plus adéquate. Comme l'illustre la Figure 35, la

réconciliation de services se compose principalement de trois couches : (i) la spécification des exigences de l'utilisateur sur les activités du processus métier, (ii) la découverte des services qui satisfont ces exigences et leur classement selon le degré de satisfaction, et enfin (iii) la sélection de services pour chacune des activités du processus métier.

La réconciliation de services se base sur les exigences de l'utilisateur. Ces dernières peuvent être de type fonctionnel et/ou non-fonctionnel. Nous parlons de « *réconciliation fonctionnelle de services* » quand il s'agit d'exigences fonctionnelles, de « *réconciliation non-fonctionnelle de services* » dans le cas d'exigences non-fonctionnelles, et de « *réconciliation de services* » quand il s'agit des deux.

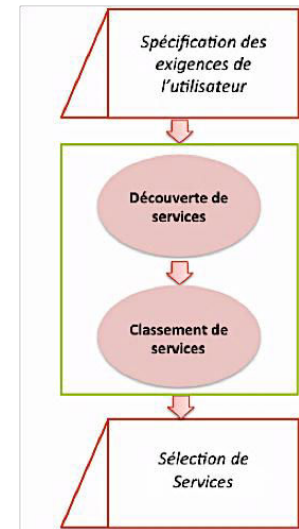


Figure 35 : Réconciliation de services.

Durant les travaux de cette thèse, nous améliorons les résultats de la réconciliation fonctionnelle établis dans la thèse de N. Boissel-Dallier [Boissel-Dallier, 2012] par l'ajout de la prise en compte des exigences non-fonctionnelles du client (cf. Figure 36). Cependant, cette amélioration prend la forme d'un deuxième filtre (qui est non-fonctionnel). Nous l'appliquons à la liste des services sélectionnés sur des critères fonctionnels dans le but de présenter au client une liste finale de services répondant à l'ensemble de ses besoins. Nous présentons dans la section qui suit le principe de la réconciliation non-fonctionnelle de services.

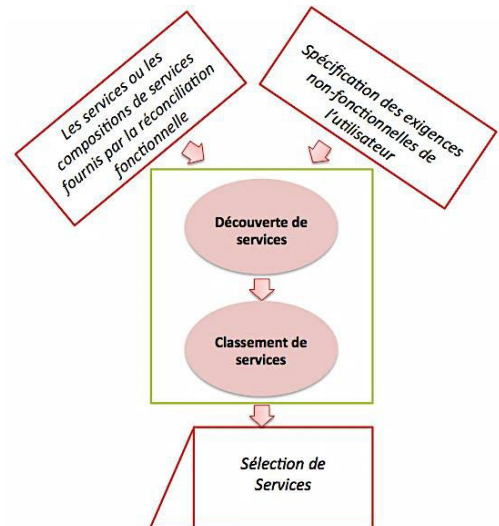


Figure 36 : Réconciliation non-fonctionnelle de services.

IV.3. Réconciliation non-fonctionnelle de services

Nous avons détaillé dans le chapitre précédent, la première couche de la réconciliation non-fonctionnelle de services, à savoir la définition des exigences non-fonctionnelles de l'utilisateur. Nous avons présenté aussi une plateforme visuelle « PetalsBPM-NFR » qui permet aux utilisateurs de modéliser et de créer leurs processus métier collaboratifs et d'annoter les activités à l'aide d'exigences non-fonctionnelles.

Nous avons aussi présenté le registre de gouvernance SOA « EasierGov-NFR » à travers lequel les fournisseurs peuvent publier leurs services enrichis par des propriétés non-fonctionnelles (via les

templates de contrats de services). Nous proposons également dans ce registre une approche de réconciliation non-fonctionnelle.

Pour la présentation de cette approche, nous considérons l'exemple d'une construction d'un pont en Lego. Il manque une pièce à ce pont (cf. Figure 37). Cette pièce est de **huit plots** (*critère fonctionnel*), de **couleur exigée grise** (*critère non-fonctionnel*), et de **nuance souhaitée Gris Fer** (*critère non-fonctionnel*). L'objectif est donc de compléter la construction du pont à partir des pièces disponibles dans la réserve de Lego.

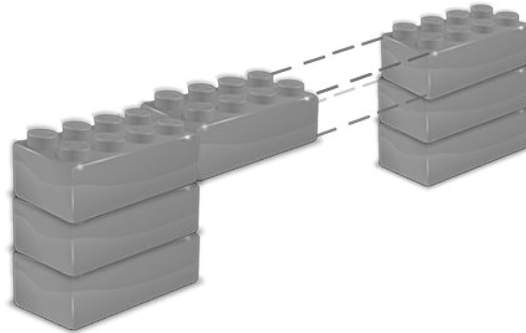


Figure 37 : Exemple d'une construction d'un pont en Lego.

Pour atteindre notre objectif, nous procédons à une recherche en deux étapes :

1) Recherche d'une pièce de huit plots

Cette recherche consiste à chercher parmi toutes les pièces à disposition, une pièce de huit plots (*critère fonctionnel*). Deux situations sont à prévoir :

- il existe une ou plusieurs pièces de Lego de huit plots. L'ensemble de ces pièces est sélectionné. Cette approche est appelée *réconciliation fonctionnelle « 1-1 »* ;
- il n'existe aucune pièce de Lego de huit plots. La difficulté est tournée par une approche de compositions de pièces (*réconciliation fonctionnelle « 1-N »*). Il s'agit d'assembler deux ou plusieurs pièces afin d'atteindre exactement, ou à défaut, de se rapprocher au plus près de la forme requise.

En effet, nous pouvons voir la réconciliation fonctionnelle « 1-N » comme une utilisation successive de la réconciliation fonctionnelle « 1-1 » sur un ensemble de combinaisons de pièces potentielles. Le temps de calcul nécessaire à l'obtention de résultats de composition est donc logiquement plus élevé que pour la réconciliation unitaire (« 1-1 »). En outre, la probabilité de proposer une composition de pièces qui réponde parfaitement au besoin risque donc d'augmenter rapidement avec l'ajout de nouvelles pièces. Par conséquent, *il est préférable de privilégier la réconciliation des plus petits ensembles possibles.*

2) Recherche d'une pièce de couleur exigée grise et de nuance souhaitée gris fer

À partir des ensembles sélectionnés à l'issue de la première recherche (filtre fonctionnel), nous effectuons une deuxième en nous basant cette fois-ci sur la couleur et la nuance requise (les deux critères non-fonctionnels).

Cette réconciliation consiste à étudier la couleur et la nuance de chacun des résultats fournis par la réconciliation fonctionnelle. Nous remarquons que dans la demande de recherche de la pièce manquante, les deux critères couleur et nuance n'ont pas le même poids : la couleur grise est *exigée* tandis que la nuance gris fer est *souhaitable*. Autrement dit, à défaut d'une pièce (ou composition de pièces) de huit plots de couleur grise et de nuance gris fer, une pièce/composition de pièces de couleur grise et de nuance qui se rapproche de celle requise et peut correspondre à l'objectif.

Nous distinguons deux cas possibles (selon le type de l'ensemble retourné par la réconciliation fonctionnelle) :

- *réconciliation non-fonctionnelle « 1-1 »* (pièces unitaires) : pour chacune des pièces fournies par la réconciliation fonctionnelle « 1-1 », nous comparons sa couleur et sa nuance avec celles requises. Dans la Figure 38, nous reprenons l'exemple présenté dans la Figure 37 afin d'illustrer la réconciliation « 1-1 ». L'ensemble sélectionné par la réconciliation fonctionnelle est composé de trois pièces de Lego de huit plots : une pièce blanche, une noire, et enfin une grise. Nous avons également à disposition les nuances de chacune : la blanche est de nuance *argile*, la noire est de nuance *noir d'ivoire*, et la grise est de nuance *gris fer*. Étant donné que la couleur grise est *exigée*, nous éliminons les pièces de couleur blanche et noire. La pièce restante est de couleur grise et de nuance *gris fer*. Ceci correspond exactement à la demande. Cette pièce est sélectionnée et le pont est construit.

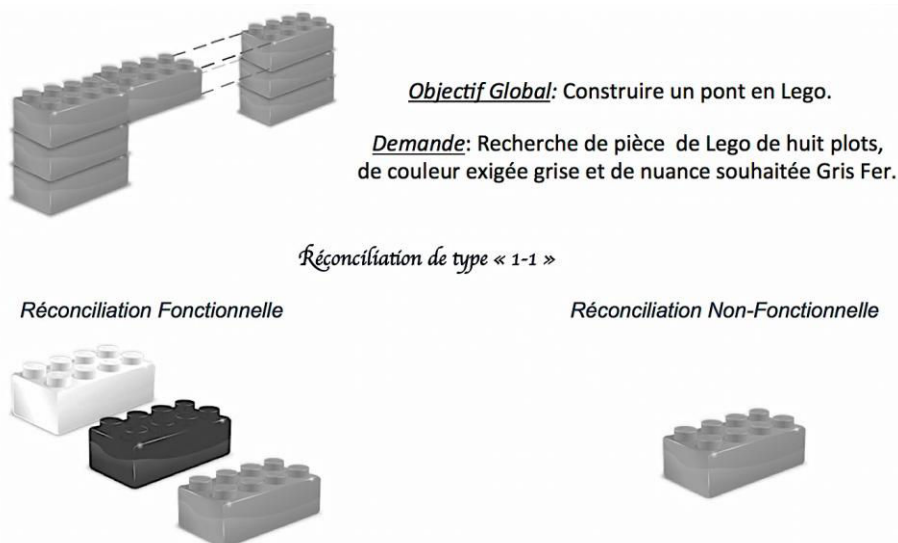


Figure 38 : Exemple d'une construction d'un pont en Lego : réconciliation « 1-1 ».

- *réconciliation non-fonctionnelle « 1-N »* (compositions de plusieurs pièces) : pour chacune des compositions fournies, nous comparons la couleur et la nuance de chacune de ces pièces par rapport à celles requises. Nous reprenons également l'exemple présenté dans la Figure 37 afin d'illustrer la réconciliation « 1-N » (cf. Figure 39). L'ensemble des pièces de Lego sélectionnées par la réconciliation fonctionnelle est constitué de trois compositions où chacune forme l'équivalent d'une pièce de huit plots. Nous remarquons que :
 - toutes les pièces de Lego utilisées dans chacune des compositions sont de couleur grise. Toutes les compositions répondent donc parfaitement à ce critère non-fonctionnel ;
 - l'absence d'une composition de pièces de nuance globale *gris Fer*. Les nuances de gris des pièces de chaque composition sont hétérogènes. Pour chacune des compositions, nous calculons le ratio « nuancePièce/nuanceRequise » de chacune de ces pièces. Par la suite, nous trions et nous classons les compositions selon leurs degrés de rapprochement de la nuance *gris Fer*.

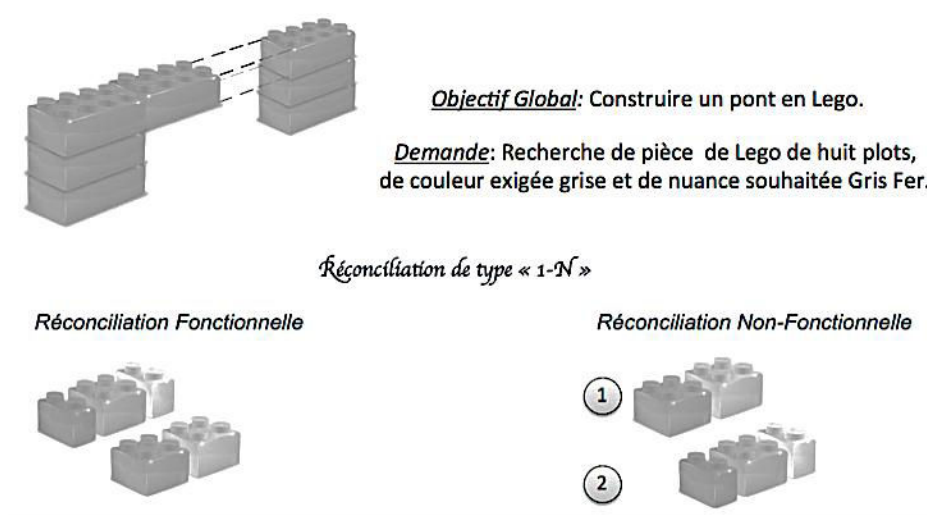


Figure 39 : Exemple d'une construction d'un pont en Lego : réconciliation « 1-N ».

Dans les sections qui suivent, nous détaillons les deux types de réconciliation non-fonctionnelle : « 1-1 » et « 1-N ».

IV.4. Réconciliation non-fonctionnelle de services de type « 1-1 »

Nous proposons dans cette section une approche pour illustrer la réconciliation non-fonctionnelle de sélection de services unitaires. Cette approche n'est nullement une nouvelle solution d'analyse multicritère. Il aurait probablement fallu en étudier plusieurs afin d'en utiliser une telle que

l'AHP (*Analytic Hierarchy Process*) [Saaty, 1988], ELECTRE (ELimination Et Choix Traduisant la REalité) [Roy, 1978], etc. Mais elles sont souvent trop figées et leur mise en œuvre est lourde pour être appliquée. Par ailleurs, nous n'avons ici pour objectif que de faire le lien entre les exigences non-fonctionnelles de l'activité métier et les propriétés non-fonctionnelles des services disponibles dans le registre de gouvernance, donc ceci n'est qu'à titre illustratif.

IV.4.1 Présentation générale de l'approche « 1-1 »

L'approche de réconciliation non-fonctionnelle de type « 1-1 » consiste à attribuer à chacune des activités du processus métier collaboratif, un service unitaire, parmi ceux qui satisfont les besoins fonctionnels, et qui répond aux exigences non-fonctionnelles (cf. Figure 40).

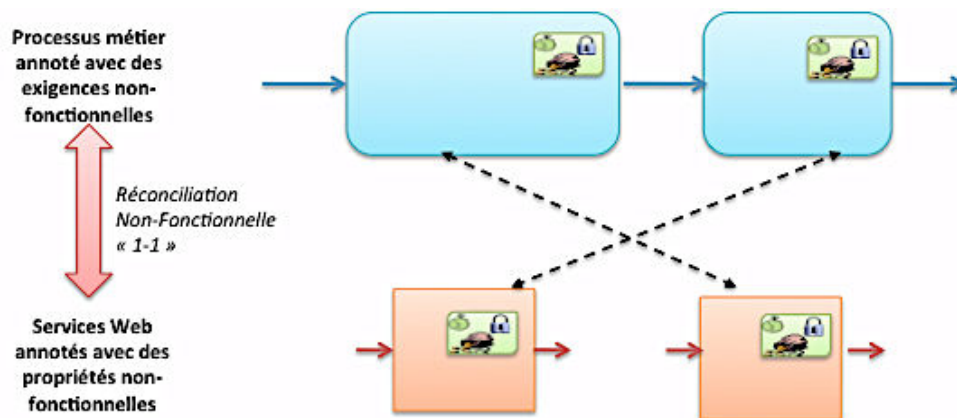


Figure 40 : Réconciliation non-fonctionnelle « 1-1 ».

IV.4.2 Algorithme de réconciliation non-fonctionnelle de type « 1-1 »

L'objectif de notre algorithme de réconciliation non-fonctionnelle est de fournir à l'utilisateur pour toute activité métier une liste triée de services Web (parmi ceux qui répondent à ses besoins fonctionnels que nous appelons services Web candidats) selon le degré de satisfaction par rapport aux exigences non-fonctionnelles.

Nous représentons les exigences non-fonctionnelles renseignées par le client pour toute activité métier, par un vecteur comme suit :

$$\text{Exigences_non-fonctionnelles} = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_n \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \end{pmatrix}$$

Où :

- P_1, P_2, \dots, P_n : l'ensemble des exigences non-fonctionnelles choisies par le client pour l'activité métier en question ;
- $\alpha_1, \alpha_2, \dots, \alpha_n$: l'ensemble des valeurs des exigences non-fonctionnelles, telles que : $\forall i \in [P_1, P_n], \alpha_i =$ la valeur de l'exigence non-fonctionnelle P_i .

Comme nous l'avons dit précédemment, nous offrons la possibilité aux clients de renseigner leurs préférences (ou pondérations) pour chacune des exigences non-fonctionnelles. Ces préférences sont également représentées par un vecteur comme suit :

$$\text{Préférences} = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_n \\ w_1 & w_2 & \dots & w_n \end{pmatrix}$$

Où :

- P_1, P_2, \dots, P_n : l'ensemble des exigences non-fonctionnelles choisies par le client pour l'activité métier en question ;
- $\omega_1, \omega_2, \dots, \omega_n$: l'ensemble des valeurs des préférences par rapport aux exigences non-fonctionnelles, telles que : $\forall i \in [P_1, P_n], \omega_i =$ la préférence de l'utilisateur par rapport à l'exigence non-fonctionnelle P_i , telle que : $\omega_i \in [0, 1]$ et $\sum_{i=1}^n (\omega_i) = 1$.

Et enfin, nous représentons l'ensemble des services Web candidats et les valeurs de leurs propriétés non-fonctionnelles choisies par l'utilisateur par une matrice comme suit :

$$\text{Services_Web_Candidats} = \begin{pmatrix} & \phi_1 & \phi_2 & \dots & \phi_n \\ S_1 & p_{11} & p_{21} & \dots & p_{n1} \\ S_2 & p_{12} & p_{22} & \dots & p_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_m & p_{1m} & p_{2m} & \dots & p_{nm} \end{pmatrix}$$

Où :

- P_1, P_2, \dots, P_n : l'ensemble des exigences non-fonctionnelles choisies par le client pour l'activité métier en question ;
- S_1, S_2, \dots, S_m : l'ensemble des services Web candidats qui satisfont les exigences fonctionnelles de l'activité métier en question.

Notre algorithme de réconciliation non-fonctionnelle se déroule en quatre étapes [Zribi et al., 2013] : (i) classification des valeurs des exigences non-fonctionnelles, (ii) élimination des services Web candidats ayant des valeurs aberrantes, (iii) normalisation et centrage des services Web candidats par rapport aux exigences non-fonctionnelles de l'activité métier et enfin (iv), classement et sélection des services Web candidats. Nous détaillons toutes ces étapes dans les sections qui suivent.

IV.4.2.1. Étape 1 : Classification des valeurs des exigences non-fonctionnelles

La première étape de notre algorithme de réconciliation non-fonctionnelle est de classer les valeurs des exigences non-fonctionnelles selon leur type. Pour l'ensemble des propriétés non-fonctionnelles de WSQF, nous distinguons trois catégories de type de valeurs : le type *textuel*, le type *booléen*, et le type *numérique* qui regroupe : *pourcentage*, *temporel*, *monétaire*, *fréquence*, et *nombre*. Dans le Tableau 6, nous présentons les différents types de valeurs des propriétés non-fonctionnelles selon le standard WSQF.

Tableau 6 : Classification des valeurs des propriétés non-fonctionnelles selon leur type.

| Textuel | Booléen | Numérique | | | | |
|---------------------|-----------------------|-------------------------|----------------------|-----------------------|----------------------|-----------------------------|
| | | Pourcentage | Temporel | Monétaire | Fréquence | Nombre |
| Message Reliability | Transaction Integrity | Availability | Business Performance | Price | Business Performance | Service Reputation |
| Controllability | Collaborability | Successability | Response Time | Penalty and Incentive | Maximum Throughput | Service Provider Reputation |
| | Informability | Accessibility | | | | Service Recognition |
| | Encryption | Standard Conformability | | | | |
| | Non Repudiation | Standard Adoptability | | | | |
| | Authentication | Relative Proofness | | | | |
| | Authorization | Observability | | | | |
| | Availability | | | | | |
| | Audit | | | | | |
| | Integrity | | | | | |
| | Privacy | | | | | |

Il est à noter que :

- nous ne considérons que les propriétés textuelles qui sont convertibles en nombre. Ces propriétés expriment un niveau d'exigence. Ce niveau peut être : très élevé, élevé, moyen, faible ou très faible. Dans ce qui suit, nous attribuons à chaque niveau une valeur. : pour le niveau très élevé, la valeur 1, 0.75 pour le niveau élevé, 0.5 pour le moyen, 0.5 pour faible et 0.05 pour le très faible ;

- les propriétés booléennes prennent deux états : soit « vrai » si valide, sinon « faux ». Dans ce qui suit, l'état vrai est représenté par « 1 » et l'état faux par « 0 » ;
- l'ensemble des propriétés de valeurs numériques peut être exprimé également par un *intervalle de valeurs numériques* (selon le choix de l'utilisateur).

Nous proposons aussi que l'ensemble des propriétés de valeurs numériques puisse être exprimé notamment par un *intervalle de valeurs numériques* (selon le choix de l'utilisateur).

Nous soulignons qu'une *propriété à maximiser de type booléen* signifie que si la valeur de la propriété est égale à « 1 » (c'est-à-dire la propriété est vraie) le maximum est atteint (par exemple le cryptage, l'authentification, etc.), contrairement à une *propriété à minimiser de type booléen* où c'est mieux quand elle est égale à « 0 ». Ce deuxième cas n'existe pas parmi les propriétés non-fonctionnelles que nous étudions, mais dans ces travaux nous souhaitons traiter l'ensemble des cas possibles.

IV.4.2.2. Étape 2 : Élimination des services candidats ayant des valeurs aberrantes

Notre objectif par cette réconciliation non-fonctionnelle est de fournir à l'utilisateur pour chacune des activités métier l'ensemble de services Web qui se rapprochent le plus de ses exigences non-fonctionnelles. Comme nous l'avons dit précédemment, nous partons d'un ensemble de services Web candidats (des services Web qui satisfont les besoins fonctionnels). Pour chacun de ces services, nous récupérons, depuis notre registre EasierGov-NFR, l'ensemble des valeurs de ses propriétés non-fonctionnelles (uniquement celles qui ont été choisies par l'utilisateur).

Cet ensemble de services Web candidats peut contenir des services Web qui satisfont suffisamment voire pleinement les exigences fonctionnelles de l'utilisateur, mais ont une ou plusieurs propriétés non-fonctionnelles qui s'éloignent totalement de ses exigences non-fonctionnelles. Les valeurs de ces propriétés non-fonctionnelles sont appelées des « valeurs aberrantes » et ces services Web ne font pas partie de ceux que nous souhaitons prendre en compte lors des calculs de réconciliation. Il convient alors de les éliminer.

Nous fournissons la possibilité aux utilisateurs d'introduire également leurs préférences par rapport à chacune des exigences non-fonctionnelles choisies. Lors de l'élimination des valeurs aberrantes, nous nous basons sur ces préférences pour déterminer le seuil de tolérance des valeurs non aberrantes (les seuils sont paramétrables dans le composant de réconciliation non-fonctionnelle) :

- $\text{Préférence} \geq 2 \times \frac{1}{\text{nombre total de propriétés choisies}}$: la préférence de l'utilisateur est d'un poids fort et le seuil de tolérance est égal à 25% ;
- $\text{Préférence} < \frac{1}{2} \times \frac{1}{\text{nombre total de propriétés choisies}}$: la préférence de l'utilisateur est d'un poids faible et le seuil de tolérance est égal à 40% ;

- Préférence non renseignée : le seuil de tolérance correspond au seuil par défaut et est égal à 30%.
- *Élimination des valeurs aberrantes dans le cas de propriété non-fonctionnelle de type numérique*

Une valeur d'une propriété non-fonctionnelle d'un service candidat est considérée comme aberrante si :

- elle est en dessous de la valeur de tolérance, dans le cas d'une propriété à maximiser ;
- elle est au dessus de la valeur de tolérance, dans le cas d'une propriété à minimiser.

Telle que : valeur de tolérance = seuil de tolérance \times valeur de la propriété dans l'activité métier

- *Élimination des valeurs aberrantes dans le cas de propriété non-fonctionnelle de type booléen ou textuel*

Une valeur d'une propriété non-fonctionnelle d'un service Web candidat est considérée comme aberrante si la préférence de l'utilisateur pour cette propriété est d'un poids fort et que la valeur de la propriété est différente de celle de l'activité métier.

- *Élimination des valeurs aberrantes dans le cas de propriété non-fonctionnelle de type intervalle de valeurs numériques*

Une valeur d'une propriété non-fonctionnelle d'un service Web candidat est considérée comme aberrante si :

- Dans le cas d'une propriété à maximiser, elle est en dessous de cette valeur :
seuil de tolérance \times valeur de la borne inférieure de la propriété dans l'activité métier
- Dans le cas d'une propriété à minimiser, elle est au dessus de cette valeur :
seuil de tolérance \times valeur de la borne supérieure de la propriété dans l'activité métier

IV.4.2.3. *Étape 3 : Normalisation et centrage des services candidats par rapport aux exigences non-fonctionnelles de l'activité métier*

Comme nous l'avons détaillé dans la première étape, les valeurs des propriétés non-fonctionnelles ne sont pas toutes de même type. En plus de cette différenciation de type des valeurs, ces propriétés n'ont pas la même dimension. Nous distinguons deux dimensions :

- Propriétés non-fonctionnelles « à maximiser » : plus la valeur de la propriété non-fonctionnelle d'un service Web est supérieure à la valeur de l'exigence non-fonctionnelle requise par l'utilisateur, mieux c'est. Par exemple, la disponibilité, les propriétés de la catégorie sécurité, la popularité du service Web;

- Propriétés non-fonctionnelles « à minimiser » : plus la valeur de la propriété non-fonctionnelle d'un service Web est inférieure à la valeur de l'exigence non-fonctionnelle requise par l'utilisateur, mieux c'est. À titre d'exemple, nous citons le temps de réponse, le prix.

Étant donné l'existence de toutes ces différenciations, les valeurs des propriétés non-fonctionnelles ne peuvent pas être comparées directement entre elles. Une étape de normalisation et de centrage devient nécessaire. Comme notre intérêt est d'être le plus proche possible des exigences de l'activité métier, nous proposons une normalisation et un centrage des valeurs des propriétés non-fonctionnelles des services Web candidats par rapport à l'activité métier. Le calcul des nouvelles valeurs normalisées et centrées des services Web candidats dépend du type de la propriété : *numérique ou textuel* (cf. Algorithme 1) ou *booléen* (cf. Algorithme 2) ou *intervalle de valeurs numériques* (cf. Algorithme 3).

L'algorithme 1 détaille les étapes de normalisation et de centrage pour les propriétés non-fonctionnelles de type *numérique* ou *textuel*.

Il prend en entrée l'ensemble des valeurs de l'activité métier (le vecteur *Exigences_non-fonctionnelles*) « E » et l'ensemble des services Web candidats ainsi que les valeurs de leurs propriétés non-fonctionnelles choisies lors de l'annotation (la matrice *Services_Web_Candidats*) « SC ». Pour chacune des valeurs de la matrice *Services_Web_Candidats*, si la propriété en question est une *propriété à maximiser* nous divisons par la valeur de la propriété correspondante dans le vecteur *Exigences_non-fonctionnelles*, puis nous soustrayons 1 au résultat de chacune (ligne 11 de l'algorithme 1). Sinon (c'est-à-dire si la propriété est une *propriété à minimiser*), nous divisons par la valeur de la propriété correspondante dans le vecteur *Exigences_non-fonctionnelles*. Puis, nous soustrayons le résultat de chacune de « 1 » (ligne 9 de l'algorithme 1). En fait, nous souhaitons que les valeurs normalisées des propriétés non-fonctionnelles des services normalisés et centrés soient positives si elles sont meilleures que les exigences demandées, nulles si identiques et négatives quand elles sont moins bonnes.

Toutefois, la valeur de l'exigence non-fonctionnelle de l'activité métier peut être égale à zéro. Dans ce cas particulier, on ne peut pas normaliser par rapport à la valeur de l'exigence. Comme nous souhaitons harmoniser les valeurs des services pour chacune des propriétés et les ramener entre 0 et 1, nous proposons donc de diviser par le plus grand du vecteur colonne de la matrice *Services_Web_Candidats* de la propriété en question (les lignes de 12 au 17 de l'algorithme 1).

Algorithme 1 : Normalisation et centrage des valeurs des propriétés des services candidats de type numérique

```

1   $E \leftarrow$  l'ensemble des exigences non-fonctionnelles de l'activité métier en question (le vecteur
2    Exigences_non-fonctionnelles)
3   $SC \leftarrow$  l'ensemble des services Web candidats et les valeurs de leurs propriétés non-
4    fonctionnelles choisies par l'utilisateur (la matrice Services_Web_Candidats)
5  Début
6    Pour  $i$  allant de 1 à  $n$  Faire :
7      Pour  $j$  allant de 1 à  $m$  Faire :
8        Pour chaque SC  $[P_i][S_j]$  Faire :
9          Si  $E[P_i] \neq 0$  Alors :
10             Si  $P_i$  est une propriété à maximiser Alors :
11                 Services_Web_Candidats_normalisés_centrés  $[P_i][S_j] = \frac{SC[P_i][S_j]}{E[P_i]} - 1$ 
12             Sinon
13                 Services_Web_Candidats_normalisés_centrés  $[P_i][S_j] = 1 - \frac{SC[P_i][S_j]}{E[P_i]}$ 
14             Fin Si.
15         Sinon
16             Max_Colonne  $[S_j] \leftarrow$  La valeur maximale du vecteur colonne  $[S_j]$  de la matrice
17             Services_Web_Candidats  $[P_i][S_j]$  correspondant à la propriété  $P_i$ 
18             Si  $P_i$  est une propriété à maximiser Alors :
19                 Services_Web_Candidats_normalisés_centrés  $[P_i][S_j] = \frac{SC[P_i][S_j]}{Max\_Colonne[S_j]} - 1$ 
20             Sinon
21                 Services_Web_Candidats_normalisés_centrés  $[P_i][S_j] = 1 - \frac{SC[P_i][S_j]}{Max\_Colonne[S_j]}$ 
22             Fin Si.
23         Fin Si.
24     Fin Pour.
25 Fin Pour.
26 Fin Pour.
27 Fin.

```

L'algorithme 2 s'intéresse aux propriétés non-fonctionnelles de type *booléen*.

Il prend en entrée l'ensemble des services Web candidats ainsi que les valeurs des propriétés non-fonctionnelles choisies lors de l'annotation (la matrice *Services_Web_Candidats*) « SC ».

Pour un booléen (par définition 0 ou 1), il n'est pas nécessaire d'effectuer une normalisation, mais uniquement d'effectuer un centrage par rapport à la valeur de la propriété en question.

Pour chacune des valeurs de la matrice *Services_Web_Candidats* et afin que la valeur normalisée et centrée soit positive si elle est mieux et négative si elle est moins bonne que l'exigence métier, il est nécessaire d'effectuer l'étape suivante :

- Dans le cas d'une *propriété à maximiser* : nous soustrayons (ligne 9 de l'algorithme 2) ;
- Dans le cas d'une *propriété à minimiser* : nous soustrayons cette valeur de « 1 » (ligne 11 de l'algorithme 2).

Algorithme 2 : Centrage des valeurs des propriétés des services candidats de type booléen

```
1  $E \leftarrow$  l'ensemble des exigences non-fonctionnelles de l'activité métier en question (le vecteur  
2   Exigences_non-fonctionnelles)  
3  $SC \leftarrow$  l'ensemble des services Web candidats et les valeurs de leurs propriétés non-fonctionnelles  
   choisies par l'utilisateur (la matrice Services_Web_Candidats)  
4 Début  
5   Pour  $i$  allant de 1 à  $n$  Faire :  
6     Pour  $j$  allant de 1 à  $m$  Faire :  
7       Pour chaque SC  $[P_i][S_j]$  Faire :  
8         Si  $P_i$  est une propriété à maximiser Alors :  
9            $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = SC [P_i][S_j] - 1$   
10          Sinon  
11             $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = 1 - SC [P_i][S_j]$   
12          Fin Si.  
13        Fin Pour.  
14      Fin Pour.  
15    Fin Pour.  
16 Fin.
```

L'algorithme 3 détaille les étapes de normalisation et de centrage pour les propriétés non-fonctionnelles dans le cas où l'utilisateur a choisi d'introduire un *intervalle de valeurs numériques*.

Dans ce cas, chacune des valeurs du vecteur *Exigences_non-fonctionnelles* est un intervalle de valeurs numériques et non une valeur simple comme dans les cas précédents. Le vecteur devient comme suit :

$$Exigences_non-fonctionnelles = \begin{matrix} & \phi_1 & \phi_2 & \dots & \phi_n \\ \left[\begin{matrix} \alpha_{1\ inf}, \alpha_{1\ sup} & \alpha_{2\ inf}, \alpha_{2\ sup} & \dots & \alpha_{n\ inf}, \alpha_{n\ sup} \end{matrix} \right] \end{matrix}$$

Cet algorithme prend en entrée l'ensemble des intervalles de valeurs numériques de l'activité métier (le vecteur *Exigences_non-fonctionnelles*) « *E* » et l'ensemble des services Web candidats ainsi que les valeurs des propriétés non-fonctionnelles choisies lors de l'annotation (la matrice *Services_Web_Candidats*) « *SC* ».

Pour chacune des valeurs de la matrice *Services_Web_Candidats*, nous distinguons trois cas possibles :

- Elle fait partie de l'intervalle de valeurs requises par l'utilisateur : elle répond à l'exigence de l'activité métier pour la propriété en question. Toutefois, il est nécessaire de la situer par rapport aux bornes de l'intervalle demandé pour savoir de quel côté elle se rapproche. Nous distinguons deux cas possibles :
 - si la propriété est une propriété à maximiser : dans ce cas plus la valeur se rapproche de la borne supérieure, mieux c'est. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 11 ;
 - si la propriété est une propriété à minimiser : dans ce cas plus la valeur se rapproche de la borne inférieure, mieux c'est. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 13 ;
- Elle est strictement supérieure à la valeur de la borne supérieure de l'intervalle :
 - si la propriété est une propriété à maximiser : dans ce cas la valeur de la propriété du service candidat est meilleure que celle demandée par l'utilisateur. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 18 ;
 - si la propriété est une propriété à minimiser : dans ce cas la valeur de la propriété du service candidat est moins bonne que celle demandée par l'utilisateur. Pour la

normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 20 ;

- Elle est strictement inférieure à la valeur de la borne inférieure de l'intervalle :
 - si la propriété est une propriété à maximiser : dans ce cas la valeur de la propriété du service candidat est moins bonne que celle demandée par l'utilisateur. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 24 ;
 - si la propriété est une propriété à minimiser : dans ce cas la valeur de la propriété du service candidat est meilleure que celle demandée par l'utilisateur. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 26.

Algorithme 3 : Normalisation et centrage des valeurs des propriétés des services candidats dans le cas d'un intervalle de valeurs numériques

```

1  $E \leftarrow$  l'ensemble des exigences non-fonctionnelles de l'activité métier en question (le vecteur Exigences_non-
2   fonctionnelles)
3  $SC \leftarrow$  l'ensemble des services Web candidats et les valeurs de leurs propriétés non-fonctionnelles choisies par
4   l'utilisateur (la matrice Services_Web_Candidats)
5 Début
6   Pour  $i$  allant de 1 à  $n$  Faire :
7     Pour  $j$  allant de 1 à  $m$  Faire :
8       Pour chaque  $S [P_i][S_j]$  Faire :
9         Si  $SC [P_i][S_j] \in E[P_{i \text{ inf}}; P_{i \text{ sup}}]$  Alors :
10          Si  $P_i$  est une propriété à maximiser Alors
11             $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = \frac{SC [P_i][S_j] - E[P_{i \text{ inf}}]}{E [P_{i \text{ sup}}] - E[P_{i \text{ inf}}]}$ 
12          Sinon
13             $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = \frac{E[P_{i \text{ inf}}] - SC [P_i][S_j]}{E [P_{i \text{ sup}}] - E[P_{i \text{ inf}}]}$ 
14          Fin Si.
15        Sinon
16          Si  $SC [P_i][S_j] < E_{i \text{ inf}}$ 
17            Si  $P_i$  est une propriété à maximiser Alors :
18               $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = \frac{SC [P_i][S_j]}{E [P_{i \text{ inf}}]} - 1$ 
19            Sinon
20               $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = 1 - \frac{SC [P_i][S_j]}{E [P_{i \text{ inf}}]}$ 
21            Fin Si.
22          Sinon
23            Si  $P_i$  est une propriété à maximiser Alors :
24               $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = -\frac{SC [P_i][S_j]}{E [P_{i \text{ sup}}]} - 1$ 
25            Sinon
26               $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = 1 - (-\frac{SC [P_i][S_j]}{E [P_{i \text{ sup}}]})$ 
27            Fin Si.
28          Fin Si.
29        Fin Si.
30      Fin Pour.
31    Fin Pour.
32  Fin Pour.
33 Fin.

```

IV.4.2.4. Étape 4 : Classement et sélection des services candidats

À l'issue de l'étape 3 de l'algorithme de réconciliation non-fonctionnelle, nous avons normalisé et centré toutes les valeurs des services Web candidats (hormis ceux qui contiennent au moins une valeur aberrante) autour des exigences de l'activité métier. En effet, chaque valeur de la matrice *Services_Web_Candidats* à l'issue de l'étape de normalisation et de centrage peut être soit :

- nulle : ceci signifie que la valeur de la propriété du service Web candidat correspond exactement à la valeur de la propriété de l'activité métier ;
- strictement positive : ceci signifie que la valeur de la propriété du service Web candidat est meilleure que la valeur de la propriété de l'activité métier ;
- strictement négative : ceci signifie que la valeur de la propriété du service Web candidat est moins bonne que la valeur de la propriété de l'activité métier.

Dans l'étape 4, nous proposons de diviser les services candidats en deux zones (cf. Figure 41) :

- Zone A : cette zone contient les services candidats qui ont des valeurs positives ou nulles pour toutes leurs propriétés non-fonctionnelles. Autrement dit, les services qui sont meilleurs que l'activité métier sur toutes les exigences non-fonctionnelles.
- Zone B : cette zone contient les services candidats qui ont au moins une propriété non-fonctionnelle de valeur négative pour toutes leurs propriétés non-fonctionnelles.

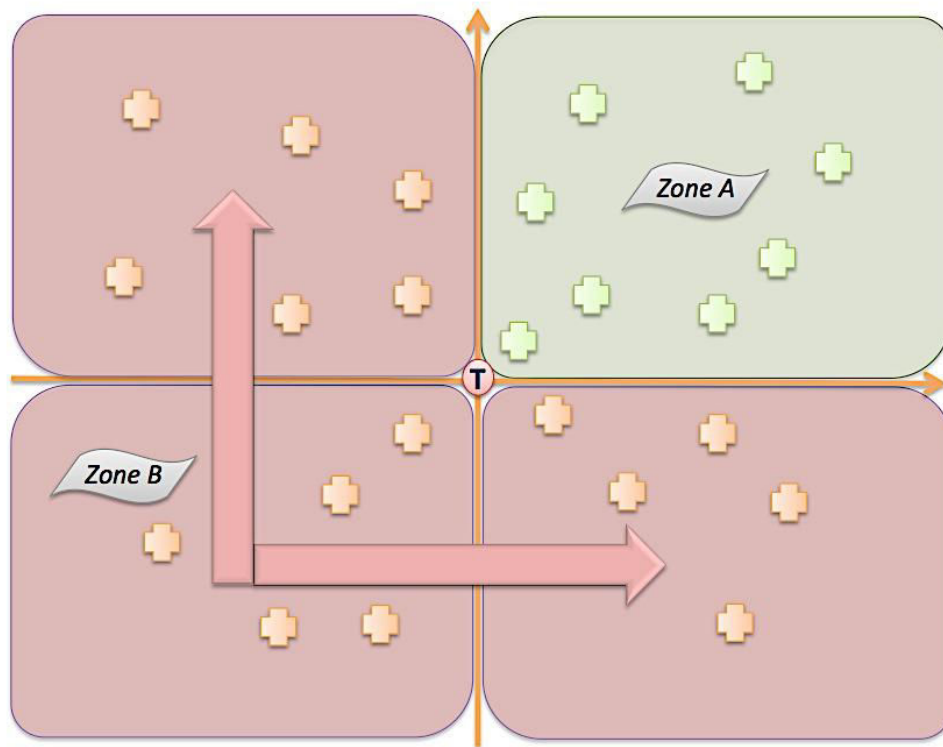


Figure 41 : Classement des services candidats dans les zones A et B.

Pour le classement et la sélection des services candidats, il est évident de commencer par classer ceux de la zone A, puis de classer à la suite les services appartenant à la zone B.

- *Dans le cas de la zone A* : classement des services Web candidats revient à calculer la distance pondérée par rapport à l'activité métier⁵. Puis les classer entre elles, le rang 1 revient à celui qui a la distance la plus grande. L'algorithme 4 détaille cette partie.
- *Dans le cas de la zone B* : cette sous-matrice comprend des services Web qui ont au moins une valeur (négative) moins bonne que celle demandée. Dans un premier temps, nous souhaitons étudier les services uniquement selon leurs valeurs défaillantes afin d'éliminer le cas où une de leurs valeurs positives récompenserait leur négativité. Pour cela, nous mettons donc à « 0 » toutes les valeurs positives des services. Tous les services vont être donc placés uniquement dans la partie négative de la zone B (cf. Figure 42).

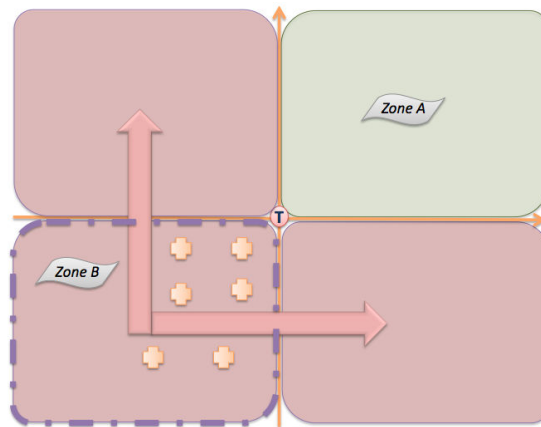


Figure 42 : Classement des services candidats de la zone B : analyse selon les valeurs négatives.

Par la suite, nous calculons la distance pondérée de chacun par rapport à l'activité métier. Le service Web candidat qui a la plus petite distance se classe en premier à la suite des services Web de la zone A.

En cas d'ex-æquo, c'est-à-dire lorsque les services Web ont exactement les mêmes moins bonnes valeurs, nous classons ces services entre eux selon les valeurs positives ou nulles initiales (les valeurs normalisées et centrées). Nous calculons la distance pondérée de ces services Web en utilisant uniquement ces valeurs. Le premier parmi eux est celui qui a la plus grande distance pondérée. L'algorithme 4 détaille cette partie.

⁵ Dans notre cas, l'activité métier se trouve à l'origine du repère dans de l'activité métier à l'issue de l'étape de normalisation et de centrage sont toutes nulles comme nous l'avons vu à l'étape précédente.

Algorithme 4 : Classement et sélection des services candidats appartenant à la zone A

```

1  $E_{n\_et\_c} \leftarrow$  l'ensemble des exigences non-fonctionnelles de l'activité métier en question après la phase de normalisation et de
2   centrage (le vecteur Exigences_non-fonctionnelles normalisé et centré)
3  $PF \leftarrow$  l'ensemble des préférences (poids) attribuées par l'utilisateur à chacune des activités métier (le vecteur Préférences)
4  $SCA_{n\_et\_c} \leftarrow$  l'ensemble des services Web candidats de la zone A et les valeurs de leurs propriétés non-fonctionnelles à 56
5   l'issue de la phase de normalisation et de centrage
6 Début
7   Pour  $i$  allant de 1 à  $n$  Faire :
8     Pour  $j$  allant de 1 à  $m$  Faire :
9       Pour chaque SCA  $[P_i][S_j]$  Faire :
10         Distance_Pondérée  $[S_j] = \sqrt{\sum_{i=1}^n (((SCA_{n\_et\_c} [P_i][S_j] - E_{n\_et\_c}[P_i])^2) \times PF[P_i])}$ 
11       Fin Pour.
12     Fin Pour.
13   Fin Pour.
14   Classement des services selon la leur valeur de distance pondérée. Le premier service est celui qui a la plus grande
15   distance.
16 Fin.

```

Algorithme 5 : Classement et sélection des services candidats appartenant à la zone B

```

1  $E_{n\_et\_c} \leftarrow$  l'ensemble des exigences non-fonctionnelles de l'activité métier en question après la phase de normalisation et de
2   centrage (le vecteur Exigences_non-fonctionnelles normalisé et centré)
3  $PF \leftarrow$  l'ensemble des préférences (poids) attribuées par l'utilisateur à chacune des activités métier (le vecteur Préférences)
4  $SCB_{n\_et\_c} \leftarrow$  l'ensemble des services Web candidats de la zone B et les valeurs de leurs propriétés non-fonctionnelles à
5   l'issue de la phase de normalisation et de centrage
6 Début
7   Pour  $i$  allant de 1 à  $n$  Faire :
8     Pour  $j$  allant de 1 à  $m$  Faire :
9       Pour chaque SCB  $[P_i][S_j]$  Faire :
10         Si  $SCB [P_i][S_j] > 0$  Alors :
11            $SCB [P_i][S_j] = 0$ 
12         Fin Si.
13       Fin Pour.
14     Pour chaque SCB  $[P_i][S_j]$  Faire :
15       Distance_Pondérée  $[S_j] = \sqrt{\sum_{i=1}^n (((SCA_{n\_et\_c} [P_i][S_j] - E_{n\_et\_c}[P_i])^2) \times PF[P_i])}$ 
16     Fin Pour.
17   Classement des services selon leur valeur de distance pondérée. Le premier service est celui qui a la plus petite
18   distance.

```

- 19 En cas d'ex-æquo de deux ou plusieurs services, nous recalculons leur distance pondérée en nous basant
 20 uniquement sur leurs valeurs positives. Ils sont classés entre eux dans le sens décroissant (le premier est celui
 21 qui a la plus grande distance).
 22 **Fin Pour.**
 23 **Fin Pour.**
 24 **Fin.**

IV.5. Réconciliation non-fonctionnelle de services de type « 1-N »

Cette section détaille notre approche de réconciliation non-fonctionnelle de services Web de type «1-N». Nous commençons, tout d'abord, par présenter le principe général de notre approche pour détailler par la suite les algorithmes de réconciliation de compositions de services Web « 1-N ».

IV.5.1.Présentation générale de l'approche « 1-N »

La réconciliation non-fonctionnelle de type « 1-N » est appelée aussi réconciliation non-fonctionnelle de compositions de services. Elle consiste à attribuer à chacune des activités du processus métier collaboratif, une composition de services, parmi celles qui satisfont les besoins fonctionnels, qui répond à ses exigences non-fonctionnelles (cf. Figure 43).

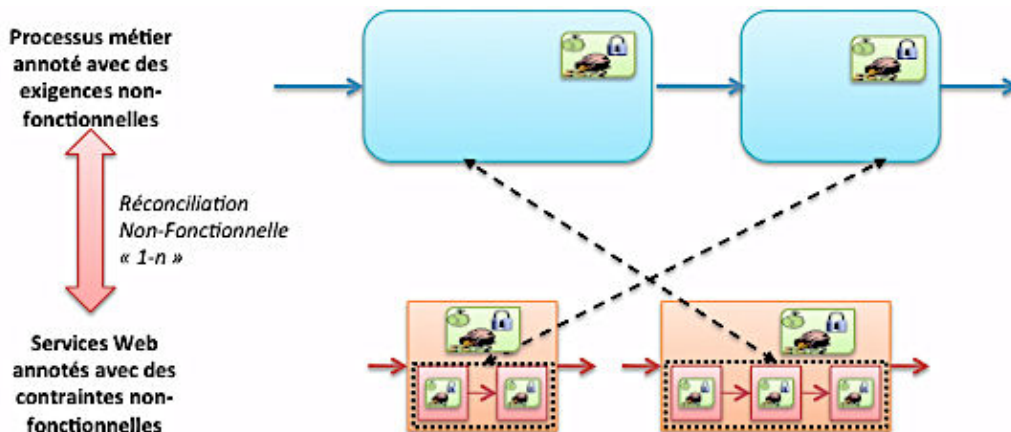


Figure 43 : Réconciliation non-fonctionnelle « 1-N ».

Nous appelons ces compositions de services « *compositions de services Web candidates* » et nous les représentons par une matrice contenant pour chacune des compositions : les valeurs des propriétés non-fonctionnelles (choisies par l'utilisateur) pour chacun de ses services Web.

$$\text{Compositions_Services_Web_Candidates} = \begin{matrix} & \phi_1 & \phi_2 & \dots & \phi_n \\ \begin{matrix} CS_1 \\ CS_2 \\ \vdots \\ CS_m \end{matrix} & \begin{pmatrix} S_1 & P_{11} & P_{21} & \dots & P_{n1} \\ S_2 & P_{12} & P_{22} & \dots & P_{n2} \\ S_3 & P_{13} & P_{23} & \dots & P_{n3} \\ S_4 & P_{14} & P_{24} & \dots & P_{n4} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_x & P_{1x} & P_{2x} & \dots & P_{nx} \\ S_y & P_{1y} & P_{2y} & \dots & P_{ny} \end{pmatrix} \end{matrix}$$

Comme nous le remarquons, l'approche de réconciliation non-fonctionnelle « 1-N » ressemble fortement à celle de la réconciliation non-fonctionnelle de type « 1-1 ». La seule différence entre ces deux est : dans la première, nous avons une matrice de compositions de services Web candidates et dans la deuxième, nous avons une matrice de services Web candidats. Toutefois, malgré cette différenciation, la forte ressemblance nous a menés à poser aux interrogations suivantes :

Pouvons nous transformer une problématique de réconciliation de type « 1-N » en une problématique de type « 1-1 » ? Si oui, comment faire ?

Cette transformation peut être possible si nous essayons de regrouper les services qui constituent la composition et de les considérer comme un seul grand service dont chacune de ses propriétés non-fonctionnelles est calculée en fonction de toutes les valeurs des services pour la propriété en question (cf. Figure 44)

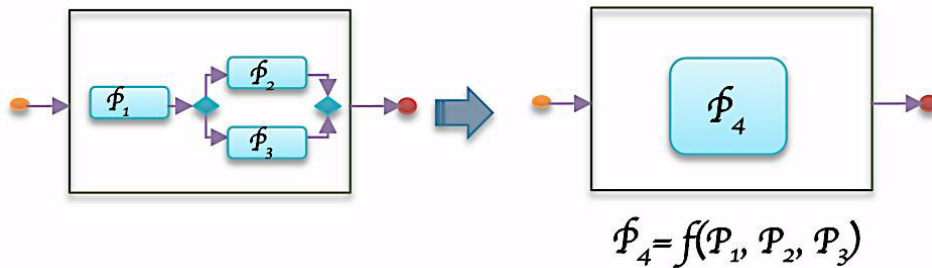


Figure 44 : Proposition de transformation d'une approche « 1-N » vers une approche « 1-1 ».

Dans ce qui suit, nous détaillons notre proposition de transformation d'une problématique de réconciliation non-fonctionnelle de compositions de services en une de sélection de services.

IV.5.2. Transformation d'une réconciliation non-fonctionnelle de type « 1-N » en une de type « 1-1 »

Le passage, d'une approche de réconciliation non-fonctionnelle de services Web en une approche de réconciliation non-fonctionnelle « 1-N », nécessite certaines transformations, comme l'illustre la Figure 45:

- \forall composition de services Web de la matrice *Compositions_Services_Web_Candidates* (CS_m) est un nouveau service Web (S_m) ;
- la valeur de chaque propriété de ce nouveau service Web (P_{nm}) doit être obtenue à partir de l'ensemble des valeurs de tous les services de la composition pour la propriété en question (elle est équivalente à la valeur de la propriété fictive de l'ensemble des services).

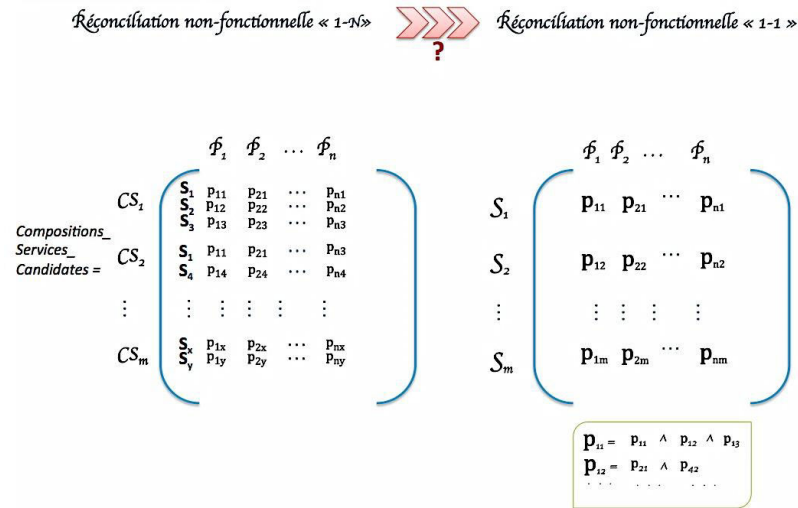


Figure 45 : Problématique de passage d'une réconciliation « 1-N » vers une de type « 1-1 ».

La résolution de cette problématique revient à déterminer les nouvelles valeurs de chaque propriété non-fonctionnelle pour chacun de ces nouveaux grands services. Pour cela, nous proposons deux étapes à suivre :

IV.5.2.1. Étape 1 : Décomposition en plusieurs sous-compositions de deux services

Les compositions de services peuvent contenir un nombre assez important de services. Afin d'analyser le déroulement de la composition en détails, nous proposons de la décomposer en une suite successives de sous-compositions qui peuvent encore se décomposer jusqu'à ce qu'obtenir dans chacune au plus deux services. Dans la Figure 46 ci-après, nous présentons un exemple de décomposition d'une composition de services.

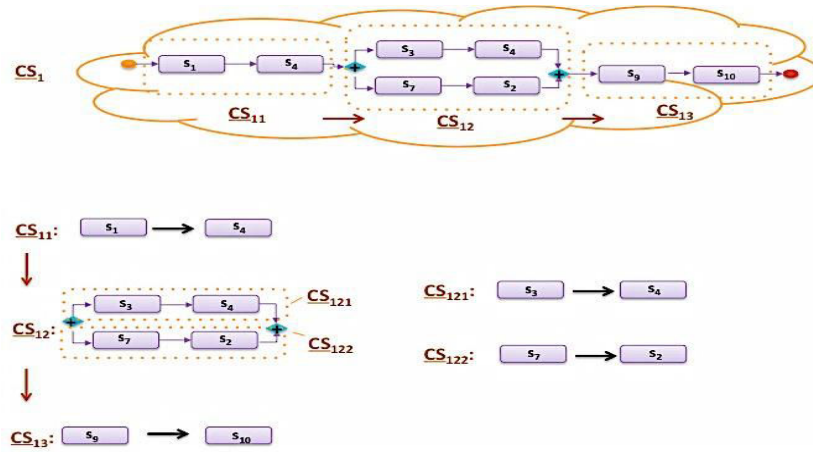


Figure 46 : Exemple de décomposition en sous-compositions de couple de services.

Nous avons commencé par décomposer la composition de services (CS_1) en trois sous-compositions (CS_{11}), (CS_{12}), et (CS_{13}). Ces trois sous-compositions de services s'exécutent séquentiellement.

- la sous-composition CS_{11} contient deux services.
- la sous-composition CS_{12} contient quatre services. Nous refaisons une autre décomposition et nous obtenons les deux sous compositions (CS_{121}) et (CS_{122}) qui s'exécutent parallèlement. Chacune des ces deux sous-compositions contient deux services. Il n'y plus donc besoin de refaire d'autres décompositions.
- la sous-composition CS_{13} contient deux services.

IV.5.2.2. Étape 2 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition

Pour calculer ces nouvelles valeurs, il faudrait prendre en considération les deux facteurs suivants : (i) le type de la propriété non-fonctionnelle (pourcentage, fréquence, temporel, monétaire ou booléen) et (ii) le type de branchement (parallèle, séquentiel, exclusif ou inclusif).

❖ Exécution des deux services

Dans le cas d'un branchement en parallèle (\diamond) (cf. Figure 47) ou d'un branchement séquentiel (\rightarrow) (cf. Figure 48), les services (S_1 et S_2) sont *exécutés tous les deux*. Par conséquent, leurs propriétés non-fonctionnelles sont *obligatoirement mises en œuvre*.

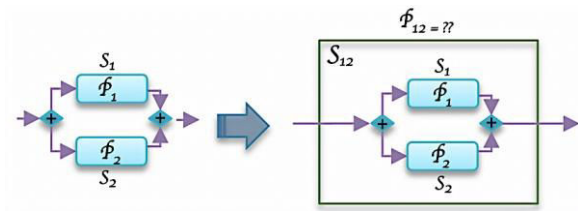


Figure 47 : Branchement en parallèle.

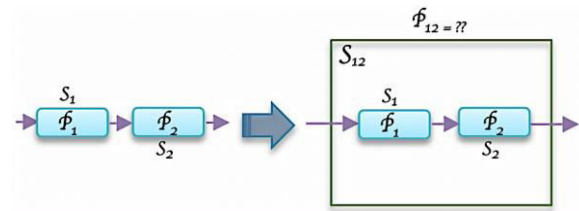




Figure 48 : Branchement séquentiel.

Pour chacune des propriétés non-fonctionnelles, la valeur globale de la propriété (P_{12}) qui correspond à l'ensemble de la sous-composition (S_{12}) est calculée comme suit :



- Temporel : dans le cas d'un branchement parallèle, étant donné que les deux services doivent s'exécuter en même temps, la valeur contraignante pour l'ensemble de la sous-composition correspond donc à la plus grande valeur des deux services. Le cas d'un branchement séquentiel où les deux services s'exécutent l'un à la suite de l'autre, il est évident que la valeur globale doit être égale à la somme des deux.
- Booléen : comme les deux services sont mis en œuvre ensemble, que ce soit pour un branchement en parallèle ou séquentiel, la validité de l'ensemble de la sous-composition ne peut être assurée que si les deux services sont valides. Dans le cas échéant, l'état de la composition est donc « faux ».
- Monétaire : du fait de l'exécution simultanée des deux services, il est évident que les deux valeurs des services sont prises en compte. La valeur monétaire de l'ensemble est égale donc à la somme des valeurs des deux services.
- Pourcentage : étant donné que les deux services doivent être activés en même temps, le pourcentage de l'ensemble de la sous-composition (que ce qui pour le branchement parallèle ou séquentiel) est équivalent au produit des pourcentages des deux services.
- Fréquence : comme les services sont portés à être activés tous les deux, la fréquence de traitement de la sous-composition globale est équivalente au minimum des fréquences des deux services. Prenons l'exemple de la propriété débit maximum : nous avons deux services S_1 et S_2 fonctionnant séquentiellement (ou parallèlement). Le premier peut fabriquer 5 bouteilles par seconde et le deuxième 10 bouchons par secondes. Le nombre de bouteilles avec bouchons par seconde issu de la sous-composition de S_1 et S_2 est donc de 5.

Le tableau suivant, Tableau 7, synthétise le calcul de la valeur de la propriété non-fonctionnelle d'une sous-composition de services dans le cas d'un branchement en parallèle et d'un branchement en série (séquentiel).

Tableau 7 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas des branchements en parallèle et séquentiel.

| | Branchement parallèle  | Branchement séquentiel  |
|--------------------------|--|---|
| Pourcentage | Produit (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) |
| Fréquence | Minimum (P_{S1}, P_{S2}) | Minimum (P_{S1}, P_{S2}) |
| Valeur Temporelle | Maximum (P_{S1}, P_{S2}) | Somme (P_{S1}, P_{S2}) |
| Valeur Booléenne | Produit (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) |
| Valeur Monétaire | Somme (P_{S1}, P_{S2}) | Somme (P_{S1}, P_{S2}) |

❖ *Exécution d'un seul service*

Dans le cas d'un branchement exclusif ( ou ) (cf. Figure 49), le flux de séquence est passé exclusivement à un seul branchement. Autrement dit, *uniquement un service* (S_1 ou S_2) qui s'exécute obligatoirement. Ce service correspond à celui qui vérifie la condition. Ceci signifie que seules les propriétés non-fonctionnelles d'un service qui s'exécute sont mises en œuvre.

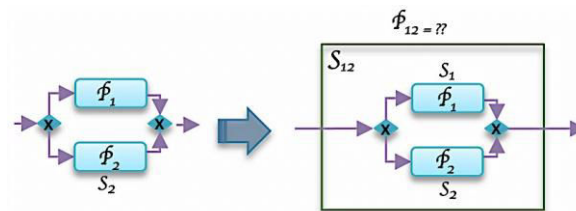




Figure 49 : Branchement exclusif.

Toutefois, nous ne pouvons connaître par avance lequel des deux services qui s'exécutera. Par conséquent, nous considérons, pour chacune des propriétés non-fonctionnelles, que la valeur de la caractéristique non-fonctionnelle globale (P_{12}) qui correspond l'ensemble de la sous-composition (S_{12}) est celle de la plus contraignante. Dans le cas où c'est une propriété à maximiser, la valeur contraignante est la plus petite des deux. Sinon (c'est-à-dire une propriété à minimiser), elle est égale à la plus grande des deux. Nous présentons dans le Tableau 8 ci-dessous la valeur de la propriété non-fonctionnelle globale de la sous-composition de services pour chacun des types des valeurs.





Tableau 8 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas d'un branchement exclusif.

| Branchement exclusif | | |
|---|------------------------------|------------------------------|
|  ou  | | |
| | Propriété à maximiser | Propriété à minimiser |
| Pourcentage | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) |
| Fréquence | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) |
| Valeur Temporelle | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) |
| Valeur Booléenne | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) |
| Valeur Monétaire | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) |

❖ *Exécution d'un seul ou des deux services*

Ce type d'exécution correspond à un branchement inclusif. Lors d'une division un ($S1$ ou $S2$) ou les deux services ($S1$ et $S2$) *sont activés*. Nous pouvons voir le branchement inclusif comme une combinaison des deux branchements parallèle et exclusif. Par ailleurs, dans ce type de branchement, on ne connaît pas par avance combien de services (et si c'est un seul lequel des deux) vont être exécutés, le calcul de la valeur de la propriété non-fonctionnelle relative à l'ensemble de la sous-composition correspond donc au pire des deux cas (entre les valeurs issues du branchement en parallèle et celles du branchement exclusif). Dans le Tableau 9 ci-dessous, nous présentons, pour chaque type de propriétés non-fonctionnelles, la valeur de la propriété globale ($P12$) qui correspond à l'ensemble de la sous-composition ($S12$).

Tableau 9 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas d'un branchement inclusif.

| | Branchement exclusif | | Branchement parallèle | Branchement inclusif | |
|-------------------|--|------------------------------|---|---|------------------------------|
| |  ou  | | |  | |
| | Propriété à maximiser | Propriété à minimiser |  | Propriété à maximiser | Propriété à minimiser |
| Pourcentage | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) |
| Fréquence | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) | Minimum (P_{S1}, P_{S2}) | Minimum (P_{S1}, P_{S2}) | Minimum (P_{S1}, P_{S2}) |
| Valeur Temporelle | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) |
| Valeur Booléenne | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) | Produit (P_{S1}, P_{S2}) |
| Valeur Monétaire | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) | Somme (P_{S1}, P_{S2}) | Minimum (P_{S1}, P_{S2}) | Maximum (P_{S1}, P_{S2}) |

Après avoir définis les règles de passage d'une problématique de réconciliation non-fonctionnelle de type « 1-N » en une de type « 1-1 », nous proposons ci-après une illustration de leur application. Nous reprenons l'exemple de la décomposition faite dans la Figure 46, et nous présentons (cf. Figure 50) le calcul de la propriété temps de réponse correspondante à la totalité de la composition de services (CS_1) dont la matrice est la suivante :

Réconciliation non-fonctionnelle « 1-N »

Temps de réponse (seconde)

$$CS_1 = \begin{pmatrix} s_1 & 3 \\ s_4 & 4 \\ s_3 & 2 \\ s_7 & 7 \\ s_2 & 5 \\ s_9 & 6 \\ s_{10} & 9 \end{pmatrix}$$

Nous détaillons dans la figure ci-dessous (cf. Figure 50), toutes les règles appliquées à chacune des sous-compositions pour le calcul du temps de réponse de la composition de services (CS_1).

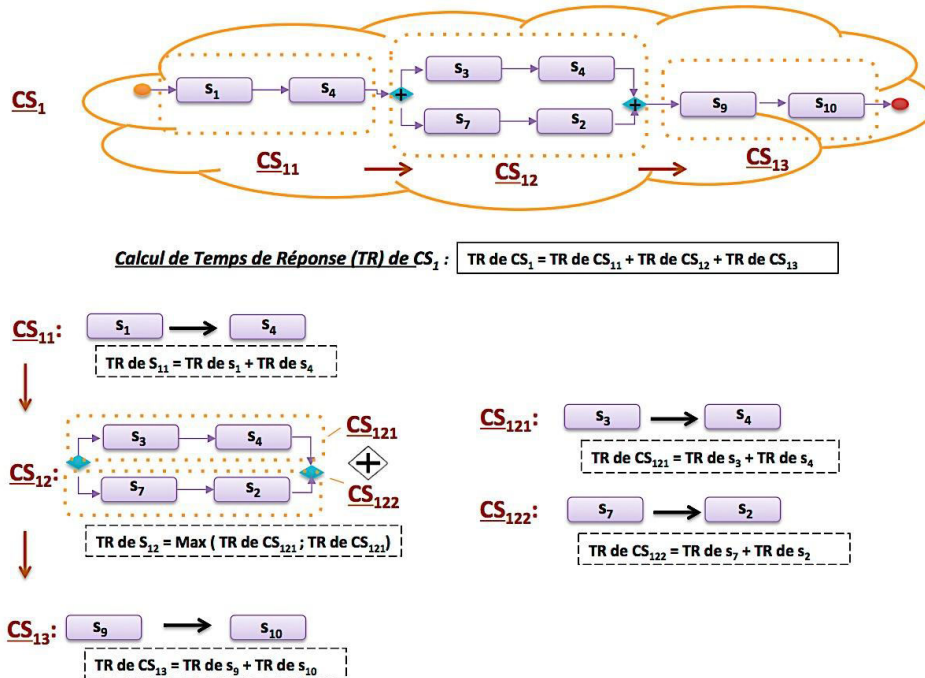


Figure 50 : Calcul du temps de réponse global d'une composition de services Web.

À partir de la matrice initiale contenant la valeur de la propriété temps de réponse pour chacun des services qui composent CS_1 et après avoir appliqué toutes les règles de calcul nous obtenons le résultat de calcul final suivant :

$$\begin{array}{c}
 \text{Réconciliation non-fonctionnelle « 1-1 »} \\
 \text{Temps de réponse global (seconde)} \\
 CS_1 = \left(\begin{array}{c} 34 \end{array} \right)
 \end{array}$$

Comme nous pouvons l'observer, nous obtenons une matrice de compositions de services candidates équivalente à celle de services candidats dans la réconciliation « 1-1 » (une seule valeur pour chaque candidat). Nous avons donc réussi à faire la transformation de la problématique « 1-N » en « 1-1 » et nous pouvons par la suite appliquer l'approche de réconciliation non-fonctionnelle « 1-1 » présentée dans la Section IV.4 de ce chapitre ou tout autre algorithme d'analyse multicritère de sélection de services.

IV.6. Conclusion

Dans ce chapitre, nous avons présenté en détails notre méthodologie de réconciliation non-fonctionnelle entre les activités du processus métier collaboratif et les services techniques satisfont les exigences fonctionnelles. Ce mécanisme de réconciliation non-fonctionnelle se divise en deux parties complémentaires selon le type de résultat retourné par la réconciliation fonctionnelle :

- La réconciliation non-fonctionnelle de services de type « 1-1 ». Cette approche recherche pour chaque activité du processus métier collaboratif, le service (parmi ceux qui satisfont les besoins fonctionnels) qui se rapproche le plus de ses exigences non-fonctionnelles. Pour cela, nous avons présenté une piste pour illustrer ce type de réconciliation.
- La réconciliation non-fonctionnelle de services de type « 1-N », appelée aussi réconciliation non-fonctionnelle de compositions de services. Cette approche s'intéresse à l'utilisation combinée de plusieurs services unitaires qui ensemble répondent au besoin de l'activité métier. Pour cela, nous proposons une approche détaillée qui permet de transformer cette problématique en une problématique de réconciliation « 1-1 ».

Une fois la réconciliation non-fonctionnelle de services est effectuée, les résultats obtenus permettent de trouver à chacune des activités métier, le ou les services qui satisfont ses besoins. Toutefois, le mécanisme proposé ici nécessite encore un certain nombre d'améliorations. Nous pouvons être confrontés au cas où aucun service et aucune composition de services ne couvre les

besoins de l'activité métier. Dans ce cas, il sera utile de combiner les attentes métier d'activités consécutives afin de trouver un ensemble de services répondant au besoin. Par ailleurs, notre mécanisme peut être amélioré en le basant sur une des méthodes de l'analyse multicritère.