Carlo Baugé
Master 2 de Recherche en Informatique
TELECOM Bretagne



Rapport de stage

Formalisation de l'évaluation des dialogueurs automatiques

Laboratoire : Département informatique de TELECOM Bretagne
Encadrants : Ioannis Kanellos, Marianne Laurent

Version : 9 août 2011



Résumé

Les dialogueurs automatiques (Spoken Dialogue System (SDS)) sont des systèmes d'interaction avec des humains au moyen du langage parlé. On peut les retrouver dans des services comme la réservation de billets d'avion, opérations bancaires, données météorologiques,... Ceux-ci offrent une interface nouvelle qui s'inscrit dans une évolution des Interfaces Homme-Machine (IHM) vers des interfaces plus naturelles qui nécessitent une nouvelle manière de penser leur évaluation. Dans le présent rapport de stage, nous commencerons par dresser un état de l'art de l'évaluation de ces systèmes, en nous appuyant sur des ouvrages et articles de référence dans l'évaluation des IHM et plus particulièrement des SDS. Nous indiquerons les principales difficultés rencontrées et les directions suivies par la recherche et l'industrie des SDS.

Cet état de l'art, les problématiques et le besoin de rationnalisation de l'évaluation que nous mettrons en évidence aboutit sur le travail de stage de formalisation de l'évaluation des SDS. Nous proposerons donc un système formel dans lequel représenter certains éléments intervenant dans l'évaluation tels que les Key Performance Indicator (KPI) qui permettent d'évaluer les caractéristiques d'un système, puis nous proposerons une représentation des différentes Communauté de Pratique (CdP) intervenant dans l'évaluation des SDS. Enfin nous verrons comment cette modélisation nous permettra de mieux comprendre certains aspects de l'évaluation.

Nous commencerons donc ce rapport par présenter les SDS, leur structure et leur fonctionnement. Nous détaillerons le contexte dans lequel ils évoluent : les IHM, pour aboutir sur la section 3 qui traite spécifiquement de l'évaluation des dialogueurs automatiques. Dans la section 3 nous traiterons des méthodes d'évaluation, les difficultés qui en ressortent et les efforts pour arriver à les résoudre. Nous rentrerons au coeur du travail de stage en introduisant le système formel dans la section 4 puis, à l'aide de ce nouveau mode de représentation nous verrons en quoi il permet d'apporter des éléments de réponse au problème de la comparaison entre différentes évaluations. Enfin nous discuterons dans la section 6 de l'importance des CdP dans l'évaluation et leur relation avec le système formel présenté.

Table des matières

1 Présentation des SDS **5**
1.1 Définition et contextualisation 5
1.2 Structure d'un SDS 6

2 L'évolution des IHM et de leur évaluation **7**
2.1 Évolution des IHM 7
2.2 Difficultés engendrées 8

3 Évaluation des SDS **8**
3.1 Approche de la recherche 11
3.2 Approche de l'industrie 11
3.3 Une volonté de convergence 12
3.4 Des mots au langage 13

4 Une représentation formelle du dialogue **14**
4.1 Opérateurs 16
4.2 Échantillons temporels de taille variable 19
4.2.1 Sous-espace vectoriel associé à un vecteur 20
4.2.2 Produit scalaire et projecteur 20
4.3 Indicateurs d'ordre supérieur 21
4.4 Application du système formel 23
4.5 Observations et limites du système formel 24

5 Comparaison de KPI **25**
5.1 Comparaison d'arbres de construction 25
5.2 Définition d'une hiérarchie 27
5.3 Mesures de similitude entre vecteurs 29

6 Caractérisation des CdP **31**
6.1 Étude syntaxique des KPI utilisés chez Orange Labs 32
6.2 Caractérisation ensembliste des CdP 33
6.3 Niveau hiérarchique associé à un KPI 34
6.4 Signature d'une CdP 35
6.5 Distance entre un KPI et une CdP 36

Références **40**

Table des figures

1 Structure séquentielle d'un SDS de type téléphonique [7] 6
2 Une définition de l'évaluation sur 3 niveaux [9] 9
3 Processus de conception d'une évaluation [9] 10
4 Relation entre Flexibilité de l'interaction et utilisabilité [14] 12
5 Exemple d'une fonction de $\mathcal{F}$ 15
6 Construction de la fonction « silence » 17

7 Opérateur dérivée « fronts montants » f^+ 18
8 Opérateur $e_k(f)$ 18
9 Exemple de projection d'un vecteur a sur $\mathcal{G}_f$ 21
10 Représentation de $p_f(a)$ dans la base $(d_i)_{i\in[\![1,n]\!]}$ de $\mathcal{F}$ (en haut) et dans la base $(e_i(f))_{i\in[\![1,ordre(f)]\!]}$ de $\mathcal{G}$ (en bas) 21
11 Représentation d'un KPI mesurant l'efficacité d'un système sous forme d'arbre (les noeuds entourés d'un rectangle sont des réels tandis que ceux entourés d'une ellipse sont des éléments de $\mathcal{F}$) 26
12 Représentation de l'arbre de construction de deux KPI – Les constructions dans $\mathcal{F}$ et dans $\mathbb{R}$ sont mises en évidence 28
13 Exemple de dialogue entre un utilisateur et un SDS donnant les informations sur les horaires de bus à Brest 31
14 analyse syntaxique d'un KPI 33
15 analyse syntaxique d'un KPI 33
16 Calcul de l'acceptation d'un nouveau KPI calculant l'efficacité, parmi les CdP d'Orange Labs 37
17 Paramètres liés aux tâches 39

Liste des tableaux

1 Opérateurs sur les éléments de $\mathcal{F}$ 16
2 Liste d'opérateurs sur les éléments de $\mathcal{F}$ 17
3 Dialogue- and communication-related interaction parameters 23
4 Distribution des niveaux hiérarchiques au sein des CdP d'Orange Labs . 35

Introduction

Depuis le 15 mars 2011 j'effectue un stage auprès du département informatique de TELECOM Bretagne. Mon travail s'inscrit dans le cadre d'une thèse menée par Marianne Laurent, dirigée par Ioannis Kanellos et financée par Orange Labs. Celle-ci a pour but d'effectuer un travail de fond sur l'évaluation des dialogueurs automatiques (Spoken Dialogue System (SDS)), la recherche d'une rationalisation et systématisation des méthodologies d'évaluation de ces systèmes. Mon travail pendant le stage se dirige donc dans cette direction et consiste à proposer une formalisation de l'évaluation de tels systèmes.

Dans un premier temps nous discuterons de l'état de l'art de l'évaluation des SDS : les différentes approches de l'évaluation ainsi que l'identification des problématiques qui en émergent et la mise en évidence de la nécessité d'une rationalisation de l'évaluation. Nous commencerons donc par introduire les SDS, leur définition et fonctionnement général pour nous pencher ensuite sur l'évaluation proprement dite. Pour cela nous partirons d'une considération générale sur l'évolution des Interfaces Homme-Machine (IHM) et de leur évaluation pour placer les SDS dans un cadre contextuel plus large. Nous analyserons ensuite l'évaluation des SDS d'un point de vue général en nous dirigeant vers les difficultés qui en émergent et les tentatives de réponse qui en découlent et qui aboutissent sur mon sujet de stage.

À la lumière de cette étude nous allons proposer une contribution à la formalisation de l'évaluation des SDS à travers la formalisation des Key Performance Indicator (KPI) et des Communauté de Pratique (CdP) qui les utilisent. Dans ce deuxième temps, nous allons donc proposer un système formel de représentation des KPI que nous étudierons, critiquerons et appliquerons à des corpus de KPI existants. À partir de là nous introduirons des outils pour comprendre et comparer les KPI entre eux. Nous nous intéresserons ensuite aux CdP en en cherchant une caractérisation formelle à partir des CdP présentes dans les projets SDS chez Orange Labs. Enfin nous associerons KPI et CdP dans une dernière partie afin de mieux les intégrer dans le contexte d'évaluation.

1 Présentation des SDS

1.1 Définition et contextualisation

Les SDS sont utilisés dans de nombreuses applications particulièrement en télécommunication. On peut penser par exemple aux services téléphoniques de réservation de billets de train. Pour ce type de services, les SDS permettent de diminuer les coûts et d'offrir une plus grande accessibilité. *Möller* [11] place les SDS dans une classe plus grande de systèmes : les systèmes intéractifs. Ceux-ci regroupent

1. les **systèmes de commande**
2. les **systèmes de dialogue par menu**
3. les **SDS**
4. les **dialogueurs multimodaux**

en les classant par ordre croissant de possibilités d'interaction. Situés à mi-chemin entre les dialogueurs multimodaux et les systèmes de dialogue par menu, les SDS ont un domaine d'application plus large que ces derniers et offrent une flexibilité plus grande mais ne se concentrent que sur le mode de communication basé sur la parole. Cependant ils gardent une large gamme de choix du degré de liberté de l'interaction et peuvent comprendre des aspects de systèmes de dialogue par menu ou de dialogueurs mulitmodaux (en tenant compte de la prosodie par exemple comme canal de communication supplémentaire). L'interaction peut être guidée par le système (à travers des questions précises par exemple) ou beaucoup plus libre en laissant l'utilisateur indiquer des informations librement sans ordre ou cadre précis.

La littérature nous propose plusieurs définitions de SDS. L'UIT-T [7] définit les SDS comme des

> « systèmes informatiques avec lesquels des utilisateurs humains interagissent à tour de rôle au moyen du langage parlé »

Ils font office d'interface entre l'utilisateur et une base de données de renseignements (horaires de train, informations météorologiques, informations bancaires,...). Le dialogue se fait à l'aide de plusieurs modules effectuant des fonctions de reconnaissance et compréhension automatique de la parole, de synthèse vocale et de gestion globale de l'interaction.

1.2 Structure d'un SDS

Il existe plusieurs architectures possibles pour intégrer ces différents modules dans le système. Une architecture possible est la structure séquentielle (Figure 1 page 6).

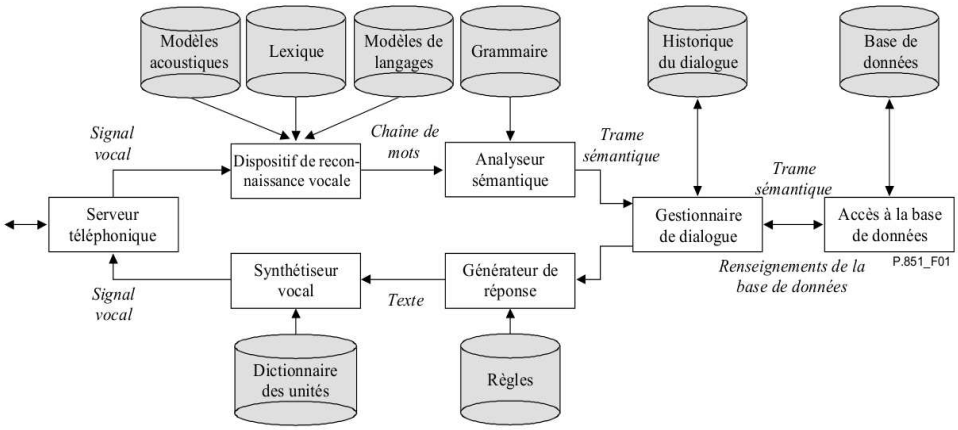


FIGURE 1 – Structure séquentielle d'un SDS de type téléphonique [7]

Une telle architecture est constituée de 6 modules principaux. À travers l'entrée fournie par l'utilisateur au serveur téléphonique, le module de reconnaissance vocale traite la voix et la décompose en une série de mots associés éventuellement à des probabilités d'erreur. L'analyseur sémantique extrait alors la structure de la phrase qui est utilisée ensuite par le gestionnaire de dialogue. Celui-ci s'assure de la cohérence générale de la conversation et extrait des informations pertinentes du contexte et de l'historique afin d'orienter

le dialogue et les décisions (demande d'informations supplémentaires, démarrage d'une procédure de correction d'erreur, demande de confirmation,...). Enfin le générateur de réponse puis le synthétiseur vocal s'occupent de contruire la réponse vocale du système.

2 L'évolution des IHM et de leur évaluation

Le travail que j'ai effectué pendant mon stage est orienté particulièrement vers l'évaluation des SDS. En effet c'est là que se concentrent aujourd'hui de nombreux problèmes centraux dans leur évolution. Ceux-ci sont critiques si l'on veut comprendre les lacunes des systèmes actuels et orienter la recherche dans ce domaine. Pour s'en convaincre il faut tout d'abord comprendre les cadres technologiques et ergonomiques nouveaux dans lesquels de tels systèmes s'inscrivent.

2.1 Évolution des IHM

D'un point de vue ergonomique, nous assistons à un profond changement dans le domaine des IHM, dans leur conception et évaluation (qui sont intimement liées). Les IHM traditionnelles proposent une interaction entre utilisateur et machine à travers des systèmes de commande physiques naturellement non ambigus (clavier, souris, bouton, ...) et donc fiables. De plus la façon dont l'interaction a lieu est déterminée principalement par la tâche et par un utilisateur dont le profil est connu, prévu. Ceci permet de définir précisément le type d'interface et de prévoir facilement le déroulement de l'interaction. De tels systèmes offrant une approche de l'interaction « traditionnelle » sont par exemple les distributeurs automatiques de billets. C'est l'utilisateur qui commence et mène l'interaction et l'échange est complètement standardisé. Aujourd'hui nous assistons à de nombreuses tendances nouvelles. *Poppe et al.* [15] en relève 4 principales :

- les **nouvelles possibilités sensorielles** offertes par de nouvelles technologies comme la reconnaissance vocale ou la détection d'expressions faciales.
- le **changement d'initiative** : de nombreux systèmes doivent être capables de prendre l'initiative et de mener, contrôler la direction de l'interaction. Ils ne doivent plus se contenter de répondre à une commande de l'utilisateur mais deviennent proactifs en étant capables de conseiller, proposer ou agir sans demande explicite de l'utilisateur.
- la **diversification des interfaces physiques** (systèmes immersifs ou à l'inverse interfaces de plus en plus petites intégrées, portables/mobiles)
- le **changement dans les objectifs des systèmes** : au lieu d'être concentrés sur la tâche à accomplir, de nouveaux systèmes s'orientent de plus en plus sur la vie de tous les jours (e.g. maisons intelligentes) et sont dirigés davantage sur l'utilisateur.

Les SDS viennent s'inscrire dans ces tendances-là même si pour certaines applications, des approches traditionnelles restent valides (si l'on se rapporte au quatrième point, on peut dire que les systèmes de réservation de billets de train par exemple restent dirigés vers la tâche à accomplir). En effet l'interface fait intervenir d'une part la parole, qui introduit simultanément une majeure liberté pour l'utilisateur et une ambiguité pour

le système. Et d'autre part le système est capable dans la plupart des cas de prendre l'initiative, suite par exemple à une abscence de réponse de la part de l'utilisateur ou dès le début pour des systèmes qui proposent une multitude de services sur la même plateforme (e.g. système HMIHY - How May I Help You - de AT&T).

2.2 Difficultés engendrées

Ces nouveautés introduisent des difficultés et des notions nouvelles à prendre en compte dans la conception et l'évaluation des SDS. *Poppe et al.* [15] en extrait six que l'on peut appliquer aux SDS.

La reconnaissance imparfaite de la parole Premièrement la communication orale est considérée plus naturelle et donc intuitivement plus efficace. Ce qui implique la présence d'un système de reconnaissance qui peut être sujet à des erreurs.

L'intégration du contexte Une deuxième notion à prendre en compte est la notion de contexte. Celle-ci est d'autant plus complexe qu'il ne semble pas y avoir de consensus sur sa définition ou les paramètres dont elle doit tenir compte.

La quantifiabilité Il existe de nombreux paramètres qui ont un impact important sur la performance perçue par l'utilisateur mais qui sont difficilement quantifiables (genre et timbre de la voix, façon de formuler les questions/réponses, ...).

Le suivi de l'apprentissage La notion d'apprentissage, prend d'autant plus d'importance que les systèmes se complexifient. Identifier les difficultés rencontrées par les utilisateurs et l'évolution de l'usage du système depuis la première utilisation est donc un besoin qui se fait sentir de plus en plus.

La limite des tests en laboratoire Enfin souvent l'évaluation de systèmes complexes tels que les SDS devrait se faire dans des conditions les plus proches possibles des conditions réelles. Cependant l'évaluation en laboratoire présente toujours des différences avec le cas d'utilisation réel.

En conclusion de cette partie on remarque que les difficultés nouvelles qui apparaissent dans la conception et l'évaluation des SDS sont dues à la prise en compte de facteurs difficilement quantifiables, subjectifs ou mal définis tels que le contexte, les intentions de l'utilisateur.

3 Évaluation des SDS

L'évaluation des SDS comme pour beaucoup de systèmes complexes est une tâche extrêmement difficile. En effet la nature même des ces systèmes nous empêche d'en avoir une vision complète. Le nombre de paramètres intervenant dans leur comportement est trop important et les liens de causalité entre ces paramètres et le système sont peu, mal ou pas connus. Nous ne possédons que quelques outils permettant de révéler certaines dimensions du système, qui nous le montrent sous un certain angle. Ils agissent de la même façon que si l'on voulait se représenter un objet à n dimensions sur une feuille de papier. Ces outils de mesure aplatissent, déforment le système réel et produisent dans le

cas des SDS des fichiers de consignation des interactions, des transcriptions de dialogues, des questionnaires de satisfaction, des données physiométriques des l'utilisateur,...

Mais ces mesures n'ont aucune valeur intrinsèque, c'est notre interprétation qui leur donne une signification et une importance. Cela dit cette valeur ne devient pas beaucoup plus grande étant donné qu'il est souvent impossible d'interpréter pertinemment directement les données brutes récoltées. Une étape supplémentaire de traitement de ces données est donc nécessaire pour les enrichir et les rendre exploitables. Dans le cas des SDS, ces traitements se réduisent souvent en pratique à des moyennes ou des calculs de pourcentages bien qu'il soit difficile ou coûteux de les calculer. Par exemple le « Nombre de demandes d'aide émanant de l'utilisateur », le « pourcentage des énoncés du système qui sont considérés comme étant pertinents dans le contexte immédiat du dialogue » [6]. On voit bien que certains traitements peuvent nécessiter l'intervention d'un humain et peuvent devenir longs et coûteux.

Les nouvelles structures résultant de ces traitements sont nommées Key Performance Indicator (KPI) en anglais, terme que nous utiliserons par la suite. Les KPI sont alors la brique de base pour la construction d'un point de vue du système et la prise de décision. Cette construction est donc un deuxième traitement des données brutes de départ. *Laurent et al.* [9] résument ce processus à trois étages à travers la figure 2 page 9.

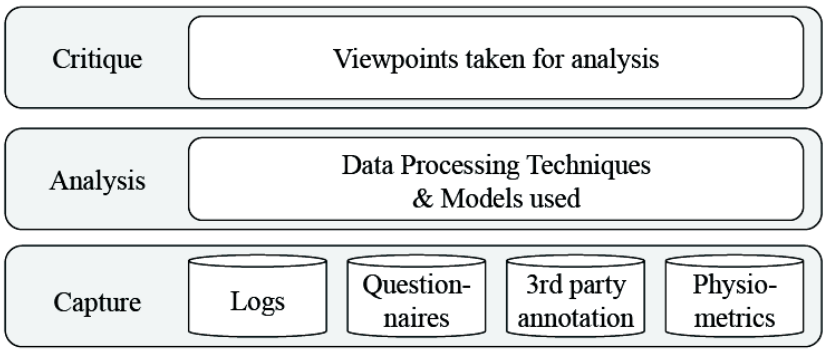


FIGURE 2 – Une définition de l'évaluation sur 3 niveaux [9]

Ce processus permet donc de créer une vision (parmi d'autres) du système qui ne retient que certains éléments, considérés comme donnant des informations pertinentes sur l'objectif recherché.

Le travail de l'évaluateur consiste à parcourir ces trois niveaux dans un sens puis dans l'autre. Il commence par définir les objectifs de l'évaluation (tels que « Est-ce que le SDS est rentable ? » ou « Est-ce que le système est capable de corriger les erreurs de compréhension ? ») qui orienteront la décision à prendre. Ces objectifs sont ensuite traduits en une série de qualités que le système doit posséder auxquelles l'évaluateur associe des KPI à calculer. Ceux-ci sont choisis en fonction de l'information qu'ils peuvent apporter sur ces qualités. Ils impliquent alors l'extraction des certains paramètres « de niveau 1 ». Grâce à cette analyse stratégique, l'évaluateur a traversé les trois niveaux en commençant par le plus haut. L'étape suivante consiste alors à mettre en application cette

stratégie, c'est-à-dire remonter les trois niveaux comme nous l'avons vu. Ce parcours en « V » est schématisé sur la figure 3 page 10.

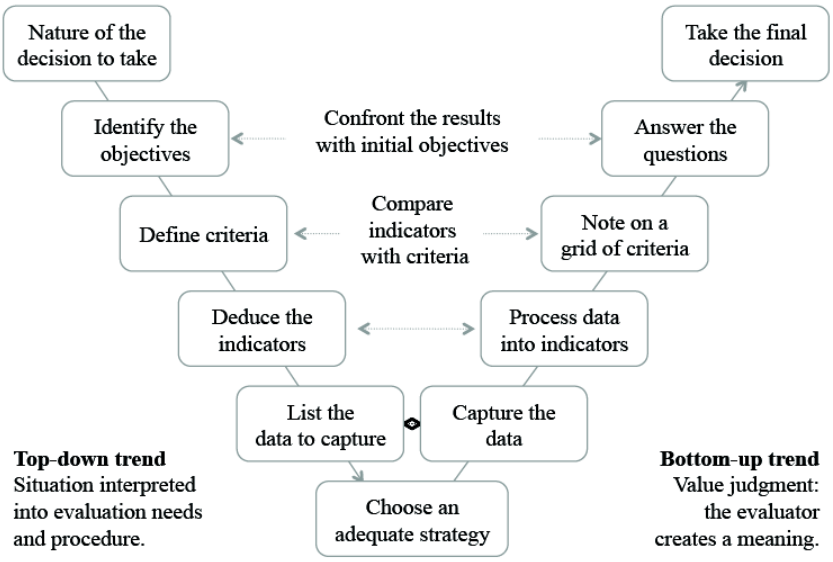


FIGURE 3 – Processus de conception d'une évaluation [9]

On se doute bien que la difficulté principale du travail de l'évaluateur provient de la première phase lorsqu'il doit chercher une stratégie d'évaluation. En effet chaque étape de traduction des objectifs généraux en qualités attendues du système puis des qualités en KPI n'est pas une tâche aisée puisqu'elle comporte un travail de choix, de sélection qui nécessite une connaissance précise du problème et du système et qui sous-entend la prise en compte d'un contexte complexe. Des questions telles que « quels points de vue du système faut-il construire compte tenu des différentes décisions à prendre à terme ? » ou « quelle est la pertinence des KPI choisis dans ce contexte donné compte tenu de ces objectifs ? » se posent alors.

Comme expliqué dans de très nombreux articles, on remarque deux tendances parallèles très différentes et par certains aspects opposées dans l'approche de conception et d'évaluation des SDS. L'une est suivie par la recherche, l'autre par les entreprises développant des systèmes commerciaux. Alors que la recherche essaie d'atteindre une grande liberté de communication et une interaction la plus naturelle possible, les systèmes commerciaux sont développés en se concentrant sur la facilité d'utilisation et la capacité à accomplir une tâche déterminée.

Ainsi les critères d'évaluation qui découlent de chacun des ces points de vue sont très différents. D'un côté la recherche s'efforce de trouver une mesure commune, qui puisse être appliquée quel que soit le service proposé ou le type d'utilisateurs auxquels on s'adresse. De l'autre l'industrie des SDS se concentre sur des recommendations et « best-practices » et des critères financiers tels que le retour sur investissement pour la conception de ses systèmes. Ainsi l'évaluation des systèmes commerciaux est différente en ce sens qu'au lieu de se demander quelle est la meilleur façon d'évaluer deux systèmes ou plusieurs versions d'un même système, la question est putôt de comment concevoir

et développer au mieux le système. Comme le remarque *Pieraccini* [13], la recherche a souvent abordé le dialogue par des principes généraux tels que ceux de Grice [4] puis, face aux limites technologiques, elle est revenue sur des stratégies de dialogue plus restrictives. A l'inverse l'industrie des SDS a commencé par une approche pragmatique où chaque interaction est conçue dans ses moindres détails, puis une fois que certaines techniques ont été maîtrisées, elle se dirige vers des types d'interaction de plus en plus libres. Si l'on se réfère à la courbe 4 page 12, on peut illustrer ces deux tendances en pensant que la recherche a attaqué la courbe du côté « like a human » alors que l'industrie l'a fait en commençant par « structured dialog ».

3.1 Approche de la recherche

Du côté de la recherche plusieurs nouvelles métriques ou méthodologies d'évaluation ont vu le jour. Certaines comme SERVQUAL, WOZ Gold Standard [12] ou SASSI [5] ce concentrent sur la recherche d'une métrique plus fiable de la qualité perçue par les utilisateurs. Alors que l'approche de PARADISE [16] essaie de trouver le lien entre certains KPI (indicateur κ, voir Figure 17 page 39 en Annexe) et la satisfaction des utilisateurs. PARADISE peut servir alors de support de comparaison entre différents systèmes, ou à prévoir la satisfaction utilisateur en fonction des indicateurs fournis par le système, ou encore à discerner les KPI du système plus indicatifs de la qualité globale. Sur ce dernier point, *T. Paek* [12] pointe certaines limites. Lorsqu'un système PARADISE est entraîné sur plusieurs SDS puis testé sur d'autres, le résultat indique que les paramètres ayant le plus d'influence sur la satisfaction des utiliasteurs sont liés à la reconnaissance vocale alors que les utilisateurs utilisent d'autres arguments pour justifier leur jugement. On voit donc que le système de PARADISE est limité quant à la comparaison de différents systèmes.

3.2 Approche de l'industrie

L'industrie tente une approche différente, plus pragmatique, dans la mesure où, consciente des limites techniques, l'effort n'est pas concentré sur le dépassement de ces limites comme pour la recherche mais sur l'optimisation des performances (ou autres critères comme le coût) compte tenu des limitations techniques.

Cette approche plus pragmatique s'applique dans le domaine de l'évaluation par une technique plus artisanale qu'on appelle de « best-practice » et qui se traduit par des recommendations, une liste de pratiques à faire ou à éviter. Le développement des applications se fait par des essais répétés jusqu'à percevoir une amélioration et la qualité des systèmes est évaluée souvent plus par l'épreuve du temps. Ces « best-practices » se sont traduites avec le temps en diverses ressources, livres, articles, séminaires sur les meilleures façon de concevoir et déployer un SDS. C'est une approche basée beaucoup sur l'expérience qui, si elle peut être justifiable, doit être validée et replacée dans son contexte (e.g. domaine d'application et population d'utilisateurs) [12].

Alors que la recherche essaie d'atteindre une grande liberté de communication et une interaction la plus naturelle possible, plus de liberté ne signifie pas toujours une meilleure

satisfaction des utilisateurs, compte tenu de la technologie actuelle. En particulier si la tâche est bien connue des utilisateurs, comme la réservation d'un billet de train : l'utilisateur sait qu'il faudra renseigner la date, la ville de départ et d'arrivée, d'éventuelles cartes de réduction, etc... Dans ces cas la restriction des libertés d'expression de l'utilisateur en choisissant un modèle d'interaction très contrôlé n'influence pas négativement l'utilisabilité et permet d'éviter des erreurs potentiellement plus fréquentes dans un modèle plus libre ([13]).

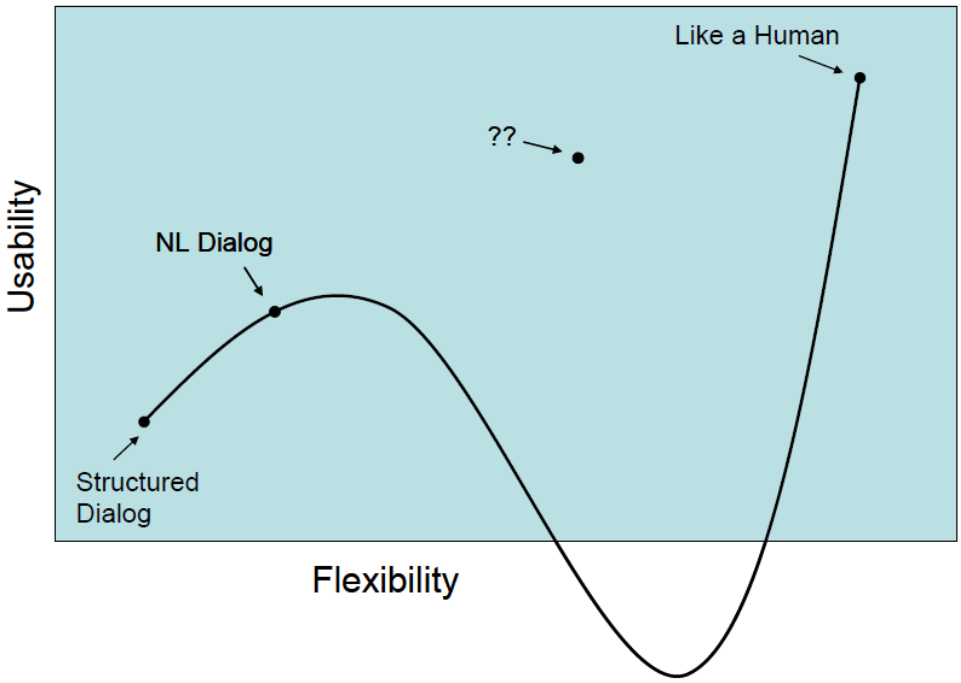


FIGURE 4 – Relation entre Flexibilité de l'interaction et utilisabilité [14]

Pieraccini et al. [14] proposent une analyse de cette relation entre utilisabilité et flexibilité (ou liberté d'expression) dans le temps (figure 4 page 12). Entre les deux extrêmes qui sont les dialogues entièrement structurés et l'interaction humain-humain la courbe n'est pas toujours croissante. Lorsqu'on s'approche de l'interaction humain-humain, plus de flexibilité n'augmente pas l'utilisabilité, au contraire. Cette courbe permet de visualiser l'endroit où les auteurs situent l'état d'aujourd'hui compte tenu des technologies actuelles. Nous nous trouvons dans un endroit où la technologie ne permet pas une intéraction entièrement libre mais permet déjà d'offrir un degré majeur de liberté d'expression. Une difficulté réside dans l'arbitrage entre liberté d'expression de l'utilisateur et guidage de l'interaction par le système.

3.3 Une volonté de convergence

Si de nombreux articles remarquent ces divergences entre recherche et industrie, tous s'accordent aussi sur la nécessité d'une convergence, d'un langage commun concernant l'évaluation. Des systèmes d'évaluation tels que PARADISE, SERVQUAL et autres

peuvent être utiles à l'industrie car le problème de commensurabilité est aussi un problème pour l'industrie. En effet les « best-practices » utilisées par l'industrie doivent être validées. À l'inverse, si l'industrie des SDS a établi un certain nombre de « best-practices » afin d'améliorer la qualité des systèmes, elle est assurée de l'optimalité de sa conception. Ainsi la recherche devrait de son côté se concentrer non seulement sur les performances relatives des systèmes mais également sur la question de comment améliorer les SDS [12].

Répondant à cette volonté de convergence et de connaissances partagées, l'UIT-T regroupe dans [6] un ensemble de KPI parmi les plus utilisés, en les groupant par catégories :

- paramètres liés au **dialogue et à la communication**
- paramètres liés à la **métacommunication**
- paramètres liés à la **coopérativité**
- paramètres liés à **l'entrée vocale**

Cette approche est en fait une taxonomie non exhaustive de points de vue possibles que l'on peut avoir sur un système. Un problème majeur reste donc même s'il est aidé par les approches vues précédemment : comment d'une part construire une vision du système qui rende compte des éléments nécessaires afin de prendre une décision ? Et comment d'autre part arriver à comparer ces multiples points de vue construits ?

C'est à ces questions que la thèse de Marianne Laurent apporte des éléments de réponse. Dans [9] par exemple *Laurent et al.* proposent une grille de lecture d'aide aux évaluateurs au cours des différentes phases de conception d'un SDS. Celle-ci permet de prendre en compte la subjectivité émanant de différentes communautés de pratiques et communautés d'intérêts, le point de vue des différents intervenants au cours du développement d'un tel système tels que les propriétaires du projet, les développeurs, les ergonomistes, les employés au marketing, les clients et les utilisateurs finaux. Il s'agit d'un squelette de réflexion à partir duquel construire une évaluation. Cet effort de rationalisation a abouti sur une plateforme implémentant cette grille de lecture (MPOWERS [8]).

Ici il ne s'agit plus de nouvelles métriques d'évaluation ni vraiment de « best-practices » mais plutôt d'une volonté de rationaliser l'évaluation, de mieux comprendre la procédure de conception d'une évaluation et de donner une structure globale et partagée entre tous les acteurs dans un projet de SDS. Nous comprenons alors qu'au-dessus de la taxonomisation et de la standardisation il est nécessaire de dessiner un cadre rationel global qui tiendrait lieu de langage commun.

3.4 Des mots au langage

Le panorama de l'évaluation aujourd'hui ressemble à une tour de Babel qui tente de trouver un langage commun. Certains mots sont utilisés de plus en plus (PARADISE, SASSI, SERVQUAL, WOZ Gold Standard,...). L'UIT-T tente de son côté d'organiser les connaissances en rassemblant et classifiant les KPI plus utilisés.

Suite à ces travaux, ce qui semble manquer n'est plus les mots mais une grammaire au langage commun, qui puisse poser les règles de base pour la construction de phrases

à partir des mots ou même de mots à partir de phonèmes. C'est-à-dire une formalisation qui définisse le passage entre les trois niveaux vus précédemment : la création de KPI à partir des données brutes et la création de points de vue à partir des KPI. En effet le problème n'est pas seulement de chercher de nouvelles métriques ou méthodologies mais d'être capable de les comprendre, de comprendre les liens entre elles et comment elles peuvent être construites.

Les résultats que nous avons abordés au cours de cet état de l'art témoignent d'une approche réductionniste au sens premier (en supprimant le caractère péjoratif qu'il a acquis de nos jours). C'est-à-dire basée sur l'hypothèse qu'un système peut être décrit et compris par l'analyse des éléments ou parties élémentaires le constituant. À partir de l'analyse de ces éléments, on effectue alors un travail inverse de reconstruction du système. C'est ce qui se passe lorsque l'évaluateur conçoit une évaluation (figure en « V »).

Le travail de recherche d'une formalisation se dirige dans une direction opposée à savoir qu'au lieu de se concentrer sur des sous-éléments du système de départ on plonge le système dans une structure d'ordre supérieur, un espace de formes qui laisse une expressivité assez grande au système pour pouvoir construire des points de vue, des caractérisations, des évaluations dont les propriétés et les relations réciproques apparaissent plus clairement.

Nous avons commencé notre réflexion à partir de deux questions importantes qui apparaissent au début de ce rapport : « Comment d'une part construire une vision du système qui rende compte des éléments nécessaires afin de prendre une décision ? Et comment d'autre part arriver a comparer ces multiples points de vue construits ? ». La première question met en évidence le fait qu'une évaluation se base avant tout sur une représentation a priori d'un système. Ce peut être une vision très complète et superficielle ou au contraire très spécifique et approfondie, mais dans tous les cas, l'évaluation ne peut être déconnectée de la façon dont on se représente l'objet à évaluer. La deuxième question s'intéresse quant à elle plutôt à la description des points de vue et donc à l'évaluation à proprement parler. Elle touche aux notions, que nous laissons volontairement vagues dans cette introduction, de « point de vue », « contexte d'évaluation ». Nous allons donc dans la suite apporter des éléments de réponse à ces deux points.

4 Une représentation formelle du dialogue

La première étape de mon travail a été de comprendre et décrire l'objet d'évaluation. Dans notre cas cet objet est le dialogue entre l'utilisateur et le système, l'output final du dialogueur automatique et non le système lui-même. En effet c'est justement notre incapacité à comprendre le système en tant que tel qui nous pousse à analyser son comportement, plus facile à comprendre et évaluer, afin d'en déduire par la suite ses propriétés, ses défauts, ses atouts. Mon travail s'est donc d'abord concentré sur une recherche d'une description, d'une représentation formelle d'un dialogue homme-machine et non d'un dialogueur automatique, d'une représentation du comportement du système et non du système en tant que tel.

La structure générale d'un système de représentation que nous avons choisi d'adopter est celle d'un langage formel constitué d'un alphabet de départ et d'un ensemble

d'opérateurs sur cet alphabet qui permettent de l'enrichir. Les éléments de cet alphabet représenteraient une valuation, un jugement ou la mesure d'une propriété élémentaire du dialogue homme-machine. Avant de vous exposer l'alphabet que nous avons choisi, réfléchissons aux qualités qu'une telle représentation doit avoir. Idéalement, elle doit être capable d'exprimer n'importe quelle caractéristique, propriété, dimension du dialogue et ce avec une précision aussi fine ou grossière que désiré. Autrement dit elle doit être flexible à la fois d'un point de vue qualitatif (possibilité d'expression de n'importe quelle propriété du dialogue) et d'un point de vue quantitatif (précision de la mesure d'une propriété). L'alphabet de base doit être partageable et compréhensible facilement entre évaluateurs et doit donc rester le plus proche possible de la perception première, directe que l'on a du dialogue.

Cette perception première du dialogue nous est fournie par une multitude de mesures sur un ou plusieurs dialogues qui prennent la forme tantôt de fichiers de log (signalant l'heure de début et fin d'un dialogue par exemple), tantôt de transcriptions, d'annotations d'experts ou de questionnaires d'utilisateurs. Cet ensemble fournit une certaine quantité de données, très hétérogène, décrivant différents aspects d'un dialogue. C'est à partir de là que pourront alors se créer des interprétations, des jugements, des points de vue sur les dialogues et donc une évaluation du système. En nous basant sur ces mesures, nous allons créer un système de représentation des propriétés d'un ou plusieurs dialogues.

L'idée sous-jacente à la représentation que nous allons proposer est la suivante : **un corpus de dialogues est une suite ordonnée et finie d'échantillons temporels auxquels on associe la présence ou non d'une propriété**. Nous allons dans un premier temps, dans un souci de clarté, considérer que ces échantillons temporels sont de durée égale et correspondent à un temps très court, le temps d'échantillonnage de la parole ou la milliseconde par exemple. Ceci est important pour que même des propriétés très spécifiques, qui s'appliquent à des temps très courts puissent être exprimées. L'ensemble est fini car il commence au début du premier dialogue par exemple et se termine à la fin du dialogue, ou à la fin d'un certain nombre de dialogues ou d'un temps prédéterminé. À l'aide de cette idée de base on peut imaginer un grand nombre de « lettres » de l'alphabet, à commencer par « l'utilisateur parle ». À chaque échantillon temporel correspond une valeur booléenne qui indique la présence ou non de la propriété « l'utilisateur parle ». La figure 5, page 15 montre une façon de représenter une telle « lettre ». Ces lettres permettent donc de représenter de manière unique toute mesure effectuée sur le système, que cette mesure vienne d'un fichier de log, d'un questionnaire de satisfaction ou autre.

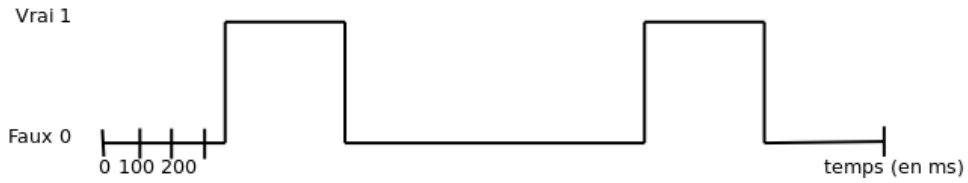


FIGURE 5 – Exemple d'une fonction de $\mathcal{F}$

Une lettre est donc une fonction de l'ensemble des échantillons temporels vers l'ensemble {0,1}. Comme l'ensemble des échantillons temporels est fini et ordonné, on peut les étiqueter entre 0 et n : $\{t_0, t_1, t_2, ..., t_n\}$. **On notera l'ensemble des échantillons temporels $\mathcal{E}$ et l'ensemble de ces fonctions $\mathcal{F}$.**

De cette façon il est facile de représenter des propriétés relativement simples du dialogue comme « l'utilisateur parle », « le système parle », « l'utilisateur énonce le mot 'horaire' », « le système comprend le mot 'horaire' », etc... Elles permettent de représenter d'une même façon n'importe quelle mesure faite sur le dialogue, indépendemment de la manière avec laquelle la mesure a été extraite. Il est donc important de noter que cette représentation ne tient pas compte de la pertinence de l'information fournie. On considère que l'information contenue dans l'alphabet de base est acceptée et partagée dans le contexte d'évaluation.

4.1 Opérateurs

Ce mode de représentation permet en théorie de représenter n'importe quelle propriété associée à un échantillon temporel. Même des propriétés plus élaborées comme « le système comprend le concept énoncé par l'utilisateur » peuvent être représentées comme des éléments de $\mathcal{F}$. Seulement une telle fonction, sous cette forme, est hermétique car elle ne donne aucune information sur ce qu'elle représente exactement. En reprenant l'exemple « le système comprend le concept énoncé par l'utilisateur », nous ne pouvons pas savoir par exemple ce qu'est un concept précisement ou sous quelles conditions un concept énoncé par l'utilisateur sera considéré comme compris par le système. Le sens de la propriété doit être clair et partagé. Or il est d'autant plus difficile de se mettre d'accord sur le sens d'une propriété lorsque celle-ci est sujette à de multiple interprétations. Ainsi si « l'utilisateur parle » est relativement consensuelle, « le système comprend le concept énoncé par l'utilisateur » l'est déjà beaucoup moins.

Nous allons dans la suite apporter des éléments de réponse à ce problème. Nous allons partir de l'hypothèse que les fonctions de $\mathcal{F}$ élaborées sont composées en réalité d'une multitude de fonctions très simples qui sont agglomérées à l'aide d'opérateurs.

Prenons un exemple. Supposons que l'on veuille calculer le taux de silence dans le dialogue. On pourrait introduire une fonction « silence » qui indique à chaque milliseconde si quelqu'un parle ou non. Cependant on remarque que cette fonction peut être aisément calculée à partir des fonctions « l'utilisateur parle » et « le système parle ». Introduisons donc deux opérateurs sur les éléments de $\mathcal{F}$ qui permettront de calculer cette fonction « silence ».

symbole	**signature**	**définition** $\forall f, g \in \mathcal{F}, \forall i \in [\![0; n]\!]$,	**remarques**
$\neg$	$\mathcal{F} \to \mathcal{F}$	$(\neg f)(t_i) = f(t_i)$	négation booléenne usuelle
$+$	$\mathcal{F} \times \mathcal{F} \to \mathcal{F}$	$(f + g)(t_i) = f(t_i) + g(t_i)$	addition booléenne usuelle

TABLE 1 – Opérateurs sur les éléments de $\mathcal{F}$

Comme l'illustre la figure 6 page 17, il est aisé de déduire la fonction « silence » des fonctions « l'utilisateur parle » (U) et « le système parle » (S) à l'aide de ces opérateurs.

Bien entendu l'exemple de la fonction « silence » reste très simple mais d'autres opérateurs peuvent être introduits. Le tableau 2 page 17 en décrit quelques uns importants classés selon leur signature.

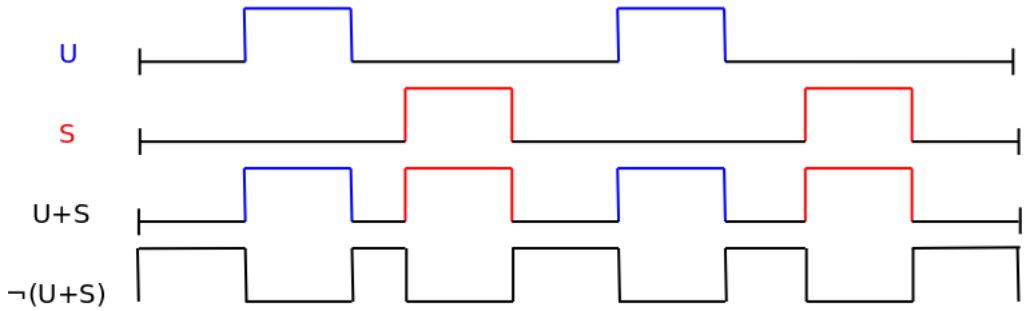


FIGURE 6 – Construction de la fonction « silence »

symbole	**signature**	**définition** $\forall f, g \in \mathcal{F}, \forall i \in [\![0, n]\!]$,
$\cdot$	$\mathcal{F} \times \mathcal{F} \to \mathcal{F}$	$(f \cdot g)(t_i) = f(t_i) \cdot g(t_i)$
$\oplus$	$\mathcal{F} \times \mathcal{F} \to \mathcal{F}$	$(f \oplus g)(t_i) = f(t_i) \oplus g(t_i)$
f_x	$\mathcal{F} \to \mathcal{F}$	$f_x(t_i) = f(t_{i-x})$
$-f$	$\mathcal{F} \to \mathcal{F}$	$-f(t_i) = f(t_{n-i})$
$e_k(f)$	$\mathcal{F} \to \mathcal{F}$	$R'' = f + R''_1$ *et* $R''(0) = 0$ $R' = \neg f \cdot R'' + R'_1$ *et* $R'(0) = 0$ $e_1(f) = R'' \oplus R'$
f^+	$\mathcal{F} \to \mathcal{F}$	$f^+ = \neg f_1 \cdot f$
f^-	$\mathcal{F} \to \mathcal{F}$	$f^- = \neg f_{-1} \cdot f$
$\int^+(f)$	$\mathcal{F} \to \{0, 1\}$	$\sum_{t \in \mathcal{E}} f(t)$
$\int^*(f)$	$\mathcal{F} \to \{0, 1\}$	$\prod_{t \in \mathcal{E}} f(t)$

TABLE 2 – Liste d'opérateurs sur les éléments de $\mathcal{F}$

Ces opérateurs permettent, à partir d'éléments de $\mathcal{F}$, d'accéder à des fonctions plus élaborées. Nous ne voulons pas dresser une liste exhaustive des opérateurs possibles puisqu'on peut en inventer une infinité et leur importance dépend de la forme des données de départ disponibles, des fichiers de log et autres mesures. Mais étudions à titre illustratif certains des opérateurs que nous avons le plus utilisés le plus souvent, en particulier lors de l'application du système aux corpus de KPI proposés dans [6] et dans les projets SDS chez Orange Labs.

L'opérateur f_x : Pour une fonction $f \in \mathcal{F}$ et $x \in [\![0, n]\!]$, f_x est la permutation circulaire de f de x échantillons temporels. Cet opérateur permet notamment de définir l'opérateur f^+.

L'opérateur f^+ : Pour une fonction $f \in \mathcal{F}$, f^+ correspond à une sorte de dérivée de f qui indique les fronts montants des « bosses » de f. Ceci est utile notamment pour repérer le nombre de fois où l'utilisateur prend la parole. Le résultat est illustré sur la figure 7 page 18. De façon analogue, l'opérateur f^- indiquera les fronts descendants de f.

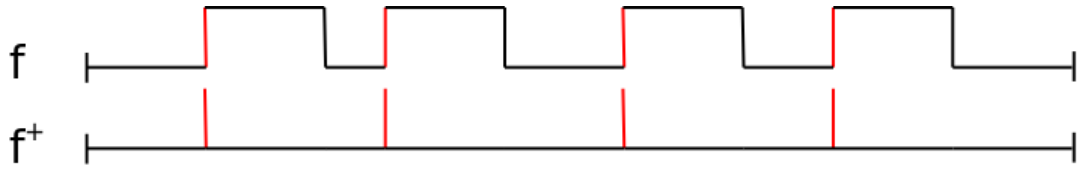


FIGURE 7 – Opérateur dérivée « fronts montants » f^+

L'opérateur $e_k(f)$: $e_k(f)$ permet d'extraire la k$^{\text{ième}}$ « bosse » de f et vaut 0 lorsque k est trop grand. Elle est utile lorsqu'on souhaite filtrer uniquement une partie du dialogue mais elle se révèlera essentielle dans la section suivante sur les échantillons temporels de taille variable. Étudions sa construction pour $k = 1$. Comme le montre le tableau 2, elle est la somme $\oplus$ de deux fonctions auxiliaires R' et R'' définies par récurrence. R" détecte le premier front montant de f, elle vaut 0 avant et 1 après le premier front montant. R' détecte le premier front descendant de f, elle vaut 0 avant et 1 après (voir figure 8).

Une fonction $g \in \mathcal{F}$ peut être définie par l'équation $g = a \cdot g_1 + b$ et $g(0) = 0$ avec $a, b \in \mathcal{F}$. Il s'agit bien d'une définition par récurrence car $\forall t \in \mathcal{E}, g(t)$ dépend de $g(t-1)$. Afin de mieux comprendre l'équation il est utile de noter que b est la condition de départ et $\neg a$ est la condition d'arrêt dans le calcul de g. Dans le tableau 2 seule $e_1(f)$ est définie. En effet il est aisé d'extraire les « bosses » suivantes à l'aide de la fonction $f \oplus e_1(f)$.

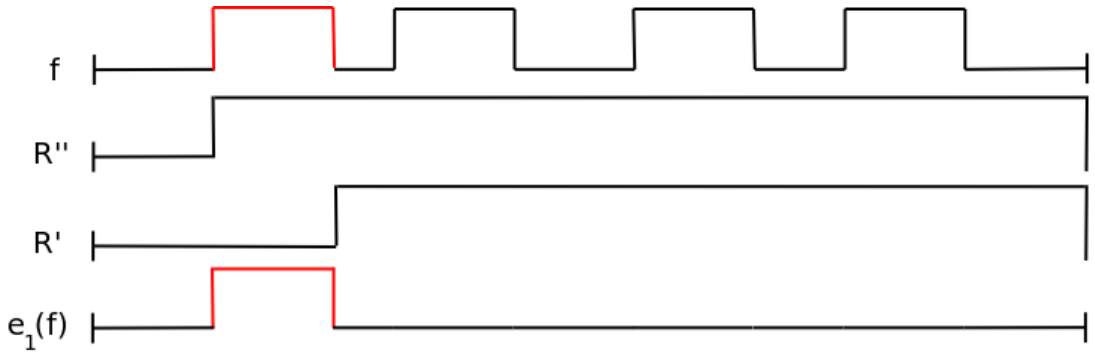


FIGURE 8 – Opérateur $e_k(f)$

L'opérateur $\int^+$: Pour une fonction $f \in \mathcal{F}$, $\int^+$ effectue la somme booléenne usuelle de tous les $f(t)$. L'opérateur $\int^*$ est analogue, au lieu d'effectuer la somme booléenne des $f(t)$, il effectue le produit booléen.

Afin de mieux saisir l'utilité de ces nouveaux opérateurs introduits, appliquons cela à un exemple. Considérons un corpus de mesures sur des dialogues de test qui représentées dans $\mathcal{F}$ nous donnent l'ensemble des propriétés de base suivantes : « l'utilisateur parle »(U), « le système parle »(S), « le mot énoncé est le mot No.i de la base de données »(Em_i) et « le mot reconnu est le mot No.i de la base de données »(Rm_i) où $i \in [\![1; nombre_de_mots]\!]$. Supposons qu'un évaluateur veuille accéder au nombre de mots correctement reconnus par le système. Les propriétés de base disponibles ne lui suffisent pas directement. Deux choix s'offrent à lui.

Soit il repère lui-même les mots bien reconnus en écoutant l'enregistrement des dialogues et en regardant les logs du système. Auquel cas, il introduit lui-même une nouvelle mesure et donc une propriété que l'on pourrait appeler « est un mot correctement reconnu par le système ». Cette nouvelle propriété est alors hermétique puisque l'on ne sait pas comment elle a été construite (sans compter les erreurs potentielles commises) et l'évaluateur devra alors convaincre les autres parties prenantes de la validité de sa mesure pour l'intégrer dans l'ensemble des propriétés de base.

Soit l'évaluateur crée la fonction « est un mot correctement reconnu par le système » à partir des propriétés de base disponibles : $\sum_{i\in[\![1;nombre_de_mots]\!]}^{\mathcal{F}}(U \cdot Em_i \cdot Rm_i)^+$. Cette fonction vaudra alors 1 à chaque fois qu'un mot est reconnu, il n'y a qu'à compter le nombre de « 1 » dans la fonction pour obtenir le nombre de mots correctement reconnus. Mais contrairement au choix précédent, cette fonction laisse transparaître, à l'aide de sa formule, sa construction de façon précise et non ambigue.

À ce stade, nous pouvons exprimer n'importe quelle propriété du dialogue à l'aide des éléments de $\mathcal{F}$ mais nous sommes désormais capables aussi d'exprimer une partie du sens attaché aux éléments de $\mathcal{F}$ plus élaborés. En effet, à l'aide d'un ensemble d'éléments de $\mathcal{F}$ simples et d'opérateurs, nous pouvons exhiber une méthode de construction ou un arbre de construction d'un élément de $\mathcal{F}$ plus élaboré. C'est cet arbre de construction ou cette formule qui est capable d'exprimer une partie du sens de la fonction élaborée de façon non ambigue. De plus cette approche permet également de réduire l'alphabet de départ qui doit être partagé entre toutes les parties. Cependant remarquons que ceci n'engage aucune unicité dans la construction et il se peut que l'on arrive à une même fonction à partir d'arbres de construction différents. Nous avons donc un système extensif puisqu'il permet d'exprimer toute propriété d'un dialogue et discernable car deux propriétés différentes auront une représentation différente.

4.2 Échantillons temporels de taille variable

Mais cet exemple nous laisse entrevoir dès à présent également un problème que nous allons traiter dans la suite. En effet les fonctions Em_i ou Rm_i représentent des propriétés sur les mots mais nous les représentons comme des propriétés associées à chaque milliseconde. Cela n'est pas dérangeant dans la mesure où un mot est constitué d'un certain nombre de millisecondes mais complique et allonge les calculs notamment en nous obligeant à utiliser l'opérateur « $^+$ » dans l'exemple précédent, dont on pourrait se passer. Nous aimerions donc trouver un moyen d'exprimer de façon minimale une propriété sur des entités plus grandes que la milliseconde, que ce soit un mot, une phrase ou un dialogue en entier. C'est-à-dire pouvoir définir différents ensembles $\mathcal{E}$ et donc différentes fonctions. Néanmoins, il ne faut pas que cela nous limite dans l'utilisation des opérateurs et nous aimerions que deux fonctions sur deux ensembles $\mathcal{E}_1$ et $\mathcal{E}_2$ différents continuent à être compatibles. Pour cela nous allons mettre en évidence certaines structures mathématiques connues présentes dans notre formalisme et nous montrerons comment elles permettent d'exprimer des propriétés de façon plus simple et minimale sur des entités plus grandes que la milliseconde.

Considérons le corps $(\{0,1\}, \oplus_{0,1}, \cdot_{0,1})$ (où $\oplus_{0,1}$ et $\cdot_{0,1}$ désignent le « ou exclusif » et le « et » booléens usuels), $(\mathcal{F}, \oplus_{\mathcal{F}})$ est un groupe commutatif qui muni de la loi externe

$\cdot$ devient un espace vectoriel sur $(\{0,1\}, \oplus_{0,1}, \cdot_{0,1})$. Nous appellerons donc désormais les éléments de $\mathcal{F}$ des vecteurs. Dès à présent nous pouvons exhiber une base naturelle de $\mathcal{F}$: $(d_i)_{i\in[\![0,n]\!]}$ où $\forall i \in [\![0,n]\!], \forall t \in [\![0,n]\!]$:

$$d_i(t) = \begin{cases} 1 & \text{si } t = i \\ 0 & \text{sinon} \end{cases}$$

C'est dans cette base que nous avons représenté nos vecteurs sur les schémas précédents. Nous allons dans la suite introduire pas à pas la notion d'échantillon temporel de taille variable.

4.2.1 Sous-espace vectoriel associé à un vecteur

Soit $f \in \mathcal{F}$ et considérons la famille de vecteurs $\forall i \in \mathbb{N}, e_i(f)$ qui représentent les « bosses » de f (c.f. tableau 2 page 17). Nous savons que les $e_i(f)$ sont nuls à partir d'un certain rang, appelons alors « ordre de f » le plus grand i tel que $e_i(f)$ soit non nul. On obtient donc la famille de vecteurs suivante :

$$(e_1, e_2, ..., e_{ordre(f)})$$

Cette famille est libre et génératrice d'un sous-espace vectoriel de $\mathcal{F}$ que nous noterons $\mathcal{G}_f$. On peut de cette manière associer à tout vecteur de $\mathcal{F}$, un sous-espace vectoriel muni d'une base, construit à partir de ses « bosses ». $\mathcal{G}_f$ est alors un sous-espace vectoriel de dimension $ordre(f)$ où chaque dimension correspond à une « bosse » de f. C'est ce genre de sous-espace vectoriel qui permettra de représenter des échantillons temporels plus grands (comme les mots, les phrases) à partir d'échantillons plus petits (comme les millisecondes).

4.2.2 Produit scalaire et projecteur

Nous allons maintenant définir un produit scalaire sur $\mathcal{F}$:

$$\forall a, b \in \mathcal{F},\ \langle a, b\rangle = \int^{+}(a \cdot b)$$

Ce produit scalaire vaudra 1 si a et b ont une partie commune et 0 sinon. On peut aisément vérifier qu'il s'agit bien d'un produit scalaire. À partir de cette définition et du paragraphe précédent nous pouvons introduire un projecteur p_f sur $\mathcal{G}_f$:

$$\forall a \in \mathcal{F},\ p_f(a) = \sum_{i\in[\![1,ordre(f)]\!]} \langle a, e_i(f)\rangle \cdot e_i(f)$$

La figure 9 page 21 illustre bien cette projection. Un vecteur a : « le système comprend le mot 'horaire' » est projeté sur chaque « bosse » de f : « l'utilisateur énonce le mot 'horaire' ». Il peut être alors représenté dans la base des $(e_i)_{i\in[\![1,ordre(f)]\!]}$, comme le montre la figure 10 page 21. Ainsi e_i représente la i-ème fois où l'utilsiateur prononce le mot « horaire ». En reprenant l'exemple des Em_i (« le mot énoncé est le mot No.i de la base de données »), la projection de ce vecteur dans le sous-espace vectoriel associé au vecteur

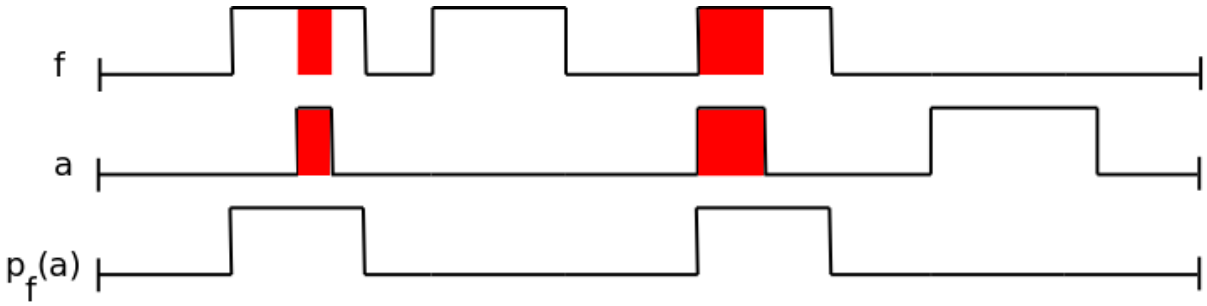


FIGURE 9 – Exemple de projection d'un vecteur a sur $\mathcal{G}_f$

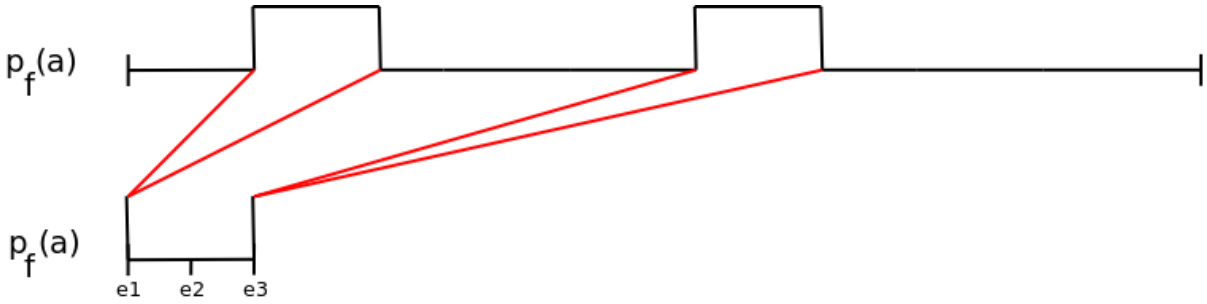


FIGURE 10 – Représentation de $p_f(a)$ dans la base $(d_i)_{i\in[\![1,n]\!]}$ de $\mathcal{F}$ (en haut) et dans la base $(e_i(f))_{i\in[\![1,ordre(f)]\!]}$ de $\mathcal{G}$ (en bas)

« est un mot » sera la fonction qui à chaque mot énoncé au cours du dialogue (et non plus chaque milliseconde) associe 1 si le mot énoncé est le mot No.i et 0 sinon.

Ainsi nous avons introduit une manière de rééchantillonner une fonction. Un échantillon temporel n'est plus obligatoirement une milliseconde ou un temps très court mais devient une « bosse », une durée plus longue et de taille variable. L'intérêt d'une telle approche sera plus clair dans les sections suivantes, cependant nous pouvons déjà avoir une idée de son utilité. En effet considérons la propriété P : « le système fournit une réponse à l'utilisateur au cours du dialogue » et sa représentation f_P dans $\mathcal{F}$. Cette propriété ne concerne pas directement une milliseconde mais un dialogue en entier. En effet l'intérêt se porte sur l'existence ou non d'une réponse de la part du système et non sur le moment auquel elle intervient. Ainsi la représentation standard dans la base des $(d_i)_{i\in[\![1,n]\!]}$ n'est pas adaptée puisqu'elle associera à chaque milliseconde d'un dialogue la même valeur : 1 si le système aura fournit une réponse et 0 sinon. En introduisant la propriété « est un dialogue » ainsi que la base associée à sa représentation dans $\mathcal{F}$, nous pouvons projeter f_P dans cette base. Nous obtenons alors une représentation adaptée où la propriété n'est plus attachée à une milliseconde mais bien à chaque dialogue.

4.3 Indicateurs d'ordre supérieur

Nous avons vu jusqu'à présent comment représenter un dialogue homme-machine, ses propriétés. Cette représentation constitue la base d'une évaluation. Elle se construit à partir d'un ensemble d'éléments de $\mathcal{F}$ qui doivent être compris, acceptés et partagés. Nous pouvons également définir des propriétés plus élaborées grâce aux opérateurs et

aux projections. Mais nous ne sommes pas encore capables d'exprimer un jugement plus compliqué sur un corpus de dialogues. Un jugement qui par exemple tient compte de multiples propriétés en les hiérarchisant, en les pondérants. C'est ce que nous allons voir dans cette nouvelle section. Comment exprimer clairement un jugement sur le dialogue à partir de cette représentation formelle dans $\mathcal{F}$.

Une évaluation ne peut être déconnectée d'un objectif, c'est-à-dire un aspect du système sur lequel on doit porter un jugement. « Le système est-il ergonomique ? » ou « Le taux de fautes de reconnaissance est-il assez bas pour une réussite des tâches de 90% ? » par exemple. L'évaluation va alors permettre de répondre de façon quantifiée à ces questions. Ce peut être une réponse binaire oui/non ou plus fine, en fournissant une note au système relativement à l'objectif considéré. Ces notes sont appelées dans la littérature Key Performance Indicator (KPI). Elle sont souvent définies en langage naturel ou par une formule mais sont le plus souvent difficilement partageables car elle sont construites pour un système en particulier et font appel à des notions ambigues.

Notre objectif désormais est d'arriver à exprimer ces KPI à l'aide du système de représentation que nous avons vu. Il faut donc réussir à passer de l'espace de description $\mathcal{F}$ à un espace tel que $\mathbb{R}$ car il doit être capable d'exprimer des quantités, des notes. Pour cela nous allons tout d'abord introduire une norme sur $\mathcal{F}$, donc une fonction de $\mathcal{F}$ vers $\mathbb{R}^+$ qui permettra d'effectuer le premier pas vers la construction d'un KPI.

Soit $\mathcal{N}$ la fonction d'un sous-espace vectoriel $\mathcal{G}$ de $\mathcal{F}$ dans $\mathbb{R}^+$ définie par :

$$\forall v \in \mathcal{G},\ \mathcal{N}(v) = \sum_{i \in [\![1; dim(\mathcal{G})]\!]} v_i$$

Où v_i est la i-ème composante du vecteur v. C'est-à-dire la fonction qui somme dans $\mathbb{R}$ les composantes du vecteur v du sous-espace vectoriel $\mathcal{G}$. Alors $\mathcal{N}$ est bien une norme. En effet elle vérifie :

la séparation :	$\forall v \in \mathcal{F},\ \mathcal{N}(v) = 0 \Rightarrow v = 0_{\mathcal{F}}$
l'homogénéité :	$\forall \lambda, v \in \{0,1\} \times \mathcal{F},\ \mathcal{N}(\lambda \cdot v) = \lvert\lambda\rvert \cdot \mathcal{N}(v)$
l'inégalité triangulaire :	$\forall u, v \in \mathcal{F}^2,\ \mathcal{N}(u+v) \leq \mathcal{N}(u) + \mathcal{N}(v)$

Cette norme permet de comptabiliser une durée, la durée d'un dialogue par exemple. Si l'on considère la fonction « est un dialogue » où l'ensemble des échantillons temporels contient les millisecondes du début à la fin d'un dialogue, la norme de cette fonction nous donnera le temps total du dialogue. Cependant dans le cas général, cette durée n'est pas forcément comptabilisée en millisecondes. En effet nous avons vu que l'on peut projeter un vecteur de $\mathcal{F}$ dans plusieurs sous-espaces vectoriels. Si l'on se place dans un de ces sous-espaces, la norme aura une unité différente de la milliseconde, plus longue, qui correspondra au sens de chaque dimension. Si l'on reprend l'exemple des figures 9 et 10, la norme de a vaudra $\mathcal{N}(a)$ dans $\mathcal{F}$ mais vaudra $\mathcal{N}(p_f(a)) = 2$ dans $\mathcal{G}_f$.

À ce stade, nous somme capables d'exprimer des KPI relativement simples qui se contentent de compter le nombre de fois où une propriété apparaît. Par exemple le temps de parole de l'utilisateur ou le nombre de questions posées au système. Pour pouvoir exprimer des concepts plus compliqués tels que le « nombre de mots par tour de parole »par exemple, il est nécessaire d'introduire des opérateurs, sur $\mathbb{R}$ cette fois, qui permettront de manipuler les normes des vecteurs de $\mathcal{F}$. Puisque nous nous retrouvons à présent dans un

espace connu, les opérateurs que nous pouvons utiliser sont tous ceux disponibles sur $\mathbb{R}$ $(+, -, /$, etc...). Cependant, en pratique, pour les corpus de KPI que nous avons utilisés, le nombre d'opérateurs sur $\mathbb{R}$ utilisés est assez restreint.

Pour résumer, pour représenter un KPI sur le système nous avons besoin d'un ensemble de vecteurs de base, d'un ensemble d'arbres de construction de vecteurs plus élaborés dans $\mathcal{F}$ et d'un arbre de construction d'un KPI dans $\mathbb{R}$.

4.4 Application du système formel

Afin de vérifier que ce système de description peut bien représenter n'importe quel KPI, nous l'avons appliqué à deux corpus de KPI : celui proposé par l'UIT-T dans [6] ainsi que le corpus des KPI (95 KPI) utilisés par les parties prenantes aux projets SDS chez Orange Labs. Ces derniers ont été obtenus par Diane Cros lors d'une étude commanditée par le laboratoire NADIA d'Orange. Nous avons donc représenté chaque KPI de ces corpus comme un nombre réel construit à partir de vecteurs de $\mathcal{F}$.

Pour ces corpus, nous avons considéré que les KPI étaient calculés à partir d'un corpus de dialogues. La colonne « Vecteurs » introduit les vecteurs de $\mathcal{F}$ nécessaires pour la construction du KPI en question et les deux dernières colonnes expriment la construction du KPI dans $\mathcal{F}$ puis dans $\mathbb{R}$. Seuls les KPI les plus représentatifs tirés de [7] sont reportés sur le tableau 3 page 23. Il s'agit de KPI représentatifs des différents types de constructions que l'on peut rencontrer. Il n'est pas utile de tous les représenter car tels qu'ils sont décrits dans [7], leur construction dépend des vecteurs de base disponibles, c'est-à-dire du système considéré et des données disponibles sur ce système. Ainsi la représentation de ces KPI n'est pas unique. Pour les KPI représentés dans le tableau 3, nous avons choisi arbitrairement certains vecteurs de base plausibles pour leur construction.

Abbr.	**Name**	**Vecteurs de base**	**Construction**	
			dans $\mathcal{F}$	dans $\mathbb{R}$
DD	dialogue duration	$f = est_un_dialogue$	f	$\frac{\mathcal{N}(f)}{\mathcal{N}(p_f(f))}$
STD	system turn duration	$U = utilisateur_parle$ $S = systeme_parle$	$f = US_1(f_1 + \neg U) + \neg SU$	$\frac{\mathcal{N}(f)}{\mathcal{N}(p_f(f))}$
SRD	system response delay	$U,\ S$	$R'' = \neg S(U^- + R''_1)$ $R' = \neg(-U)(R'_1 + (-S)^-)$ $R = -R' \cdot R'' \cdot \neg(U + S)$	$\frac{\mathcal{N}(R)}{\mathcal{N}(p_R(R))}$
PA :CO	number of correctly parsed user utterances	$UC_i = U_enonce_concept_i$ $SC_i = S_comprend_concept_i$	$f = \prod_i \neg(p_U(UC_i) \oplus p_U(SC_i))$	$\mathcal{N}(f)$

Table 3 – Dialogue- and communication-related interaction parameters

4.5 Observations et limites du système formel

Cette forme de représentation semble pouvoir bien exprimer toute propriété d'un dialogue avec un degré de précision variable à souhait. Nous avons pu l'appliquer sur des KPI issus de la littérature ainsi que sur un corpus de KPI spécifiques aux projets SDS chez Orange Labs. Cette forme de représentation permet un partage plus facile et une meilleure compréhension des différents KPI entre différentes parties d'un projet d'évaluation de SDS. En effet le problème du partage et de la comparaison de points de vue d'évaluation ont été mis en évidence dans l'état de l'art. Cependant en aucun cas ce mode de représentation ne permet de juger un KPI ou comparer autrement que de façon purement descriptive deux KPI. Mais signalons les limites du système que nous avons identifiées.

Tout d'abord il ne permet de représenter que des propriétés, ce qui veut dire que si l'on veut représenter des caractéristiques discrètes du dialogue telles que « identifiant (dans la base de données) du mot reconnu par le système », il nous faut « découper » celles-ci en un ensemble de propriétés : « le mot reconnu par le système a l'identifiant i dans la base de données ». De même, pour les caractérstiques continues telles que « charge du serveur vocal », il faudra dans un premier temps les discrétiser (à l'aide d'intervalles de valeurs par exemple), puis les transformer comme précédemment en un ensemble de propriétés.

Deuxièmement, bien que nous n'ayons pas rencontré dans la littérature de tels KPI, nous pouvons imaginer des KPI plausibles mais difficilement représentables dans notre système tels que « nombre de dialogues où le taux de reconnaissance vocale est supérieur à 80% ». En effet nous avons besoin de construire (s'il n'est pas déjà construit) le vecteur v : « est un dialogue avec un taux de reconnaissance vocale supérieur à 80% », projeté dans le sous-espace associé à « est un dialogue ». Le taux de reconnaissance vocale est lui-même un KPI. Il faut donc construire pour chaque dialogue le KPI « taux de reconnaissance vocale »(nombre de mots bien reconnnus sur nombre de mots total) puis utiliser les résultats pour chaque dialogue pour construire le vecteur v. Ainsi, contrairement à ce que nous avons vu jusqu'à présent, ce KPI plus compliqué n'est pas uniquement un ensemble de vecteurs de $\mathcal{F}$ utilisés ensuite pour construire un KPI dans $\mathbb{R}$, mais il rajoute un autre niveau en réutilisant des KPI pour construire de nouveaux vecteurs de $\mathcal{F}$ et enfin construire le résultat final dans $\mathbb{R}$.

Tous les KPI que nous avons utilisés (aussi bien pour l'UIT-T que pour Orange Labs) sont décrits sans indiquer les vecteurs de base nécéssaires mais en indiquant uniquement le mode de construction. Ainsi pour la plupart des KPI, une ambiguité réside et nous oblige à faire des choix arbitraires. Dans le tableau 3, la colonne « Vecteurs » ne provient donc pas directement de la définition du KPI tel que décrit dans la littérature mais est devinée à partir de la définition et pourra être légèrement différente d'un système à l'autre, selon les données disponibles. Par exemple pour le KPI $PA : CO$ (number of currectly parsed user utterances) du tableau 3 nous avons choisi une formule construite à partir des vecteurs « U énonce le concept No.i » et « S comprend le concept No.i » mais si ces vecteurs ne sont pas disponibles directement on pourra être obligé de reconstruire ces vecteurs à partir des vecteurs disponibles.

D'autre part, lorsqu'un vecteur est représenté dans un espace inadapté comme par exemple le vecteur « le système comprend le mot 'horaire' » représenté dans l'espace des

millisecondes, est-ce que la fonction vaut 1 pendant toute la durée d'énonciation du mot « horaire » ou est-ce qu'elle vaut 1 uniquement au moment où le système identifie le mot « horaire » ? Ce choix n'est pas critique mais il peut mener à des ambiguités s'il n'est pas défini clairement.

Pour ce qui concerne la projection dans un sous-espace associé à un vecteur tel que « est un mot » il est nécessaire d'imposer une contrainte sur la forme du vecteur « est un mot » car cela n'a de sens que si chaque « bosse » de ce vecteur correspond à un mot prononcé. Autrement il ne faut pas que deux mots se suivent sans pause (et se retrouvent dans la même bosse) et quitte à l'introduire artificiellement, il faut s'assurer qu'il existe au moins un échantillon temporel séparant deux mots consécutifs.

Enfin le produit scalaire que nous utilisons contraint la projection. En effet si l'on considère le vecteur v « le système comprend le mot 'horaire' » et que l'on veut le projeter sur le sous-espace associé à p « est une phrase », dans la base associée à p la composante i de v vaudra 1 si le mot « horaire » a été prononcé **au moins une fois** au cours de la i-ème phrase. Ce « au moins » est imposé par le choix du produit scalaire. Mais on pourrait avoir besoin d'autres types de produits scalaires traduisant les notions de « au plus n fois » ou autre. Dans ce cas il faudra changer de produit scalaire.

5 Comparaison de KPI

Comme le montre [10] ou [9], le processus d'évaluation d'un SDS implique de nombreuses CdP, chacune ayant sa propre vision du système à évaluer. Les maîtres d'oeuvre, les experts techniques, les ergonomes, le client, les personnes du marketing sont autant de parties qui ne s'intéressent qu'à certains aspects bien spécifiques du système. Si théoriquement nous sommes capables d'exprimer le point de vue de chacun, comment pouvons-nous les comparer entre eux ? Il ne s'agit pas de les classer à travers une méta-évaluation mais bien de comprendre leurs différences. Nous souhaitons maintenant comprendre les mécanismes de construction de KPI pour discerner leurs points communs et différences et expliquer leurs choix. Nous allons donc à la fois chercher à comparer les KPI entre eux et chercher à caractériser les différentes CdP.

La comparaison des KPI va consister à repérer et quantifier leurs différences. Celles-ci peuvent intervenir à plusieurs endroits : au niveau de l'ensemble des vecteurs de base, au niveau des arbres de construction dans $\mathcal{F}$ ou au niveau de l'arbre de construction dans $\mathbb{R}$. Nous allons tout d'abord nous intéresser à la comparaison des arbres de construction puis nous verrons comment comparer des vecteurs de base différents.

5.1 Comparaison d'arbres de construction

Un KPI se construit à partir d'un objectif d'évaluation. Par exemple « le système est-il efficace ? », « est-il rentable ? ». Celui-ci est ensuite traduit de façon de plus en plus précise en décrivant chaque partie ou sous-objectif. Que veut dire « efficace », à quels paramètres cela fait référence ? En prenant la définition de l'UIT-T [6], l'efficacité rassemble les « mesures de la précision et de la complétude des tâches système relatives

aux ressources (temps, effort humain, ...) utilisées pour exécuter les différentes tâches du système ». En précisant les termes on en vient ensuite à des formules mathématiques comme illustré sur la figure 11 page 26. Au premier niveau de l'arbre, nous avons traduit le terme « relatives » de la définition d'efficacité par l'opérateur « / » sur $\mathbb{R}$. Puis la « quantité de ressources utilisées » est traduite comme la somme de la durée du dialogue avec une mesure de l'effort humain (fournie par exemple par un questionnaire utilisateur). Les KPI entourés en bleu clair, c'est-à-dire ceux à la frontière entre $\mathbb{R}$ et $\mathcal{F}$ sont les plus simples, ils ne sont calculés qu'à l'aide de la norme d'un vecteur. La précision continue lorsqu'on passe de $\mathbb{R}$ à $\mathcal{F}$ où l'on exprime la construction des vecteurs de $\mathcal{F}$ utilisés en écrivant leur formule. Enfin aux feuilles de l'arbre nous trouvons alors les vecteurs de base nécessaires à la construction du KPI. C'est cette démarche « top-down » depuis l'objectif jusqu'aux vecteurs de $\mathcal{F}$ qu'effectue l'évaluateur afin de trouver le KPI qui lui convient.

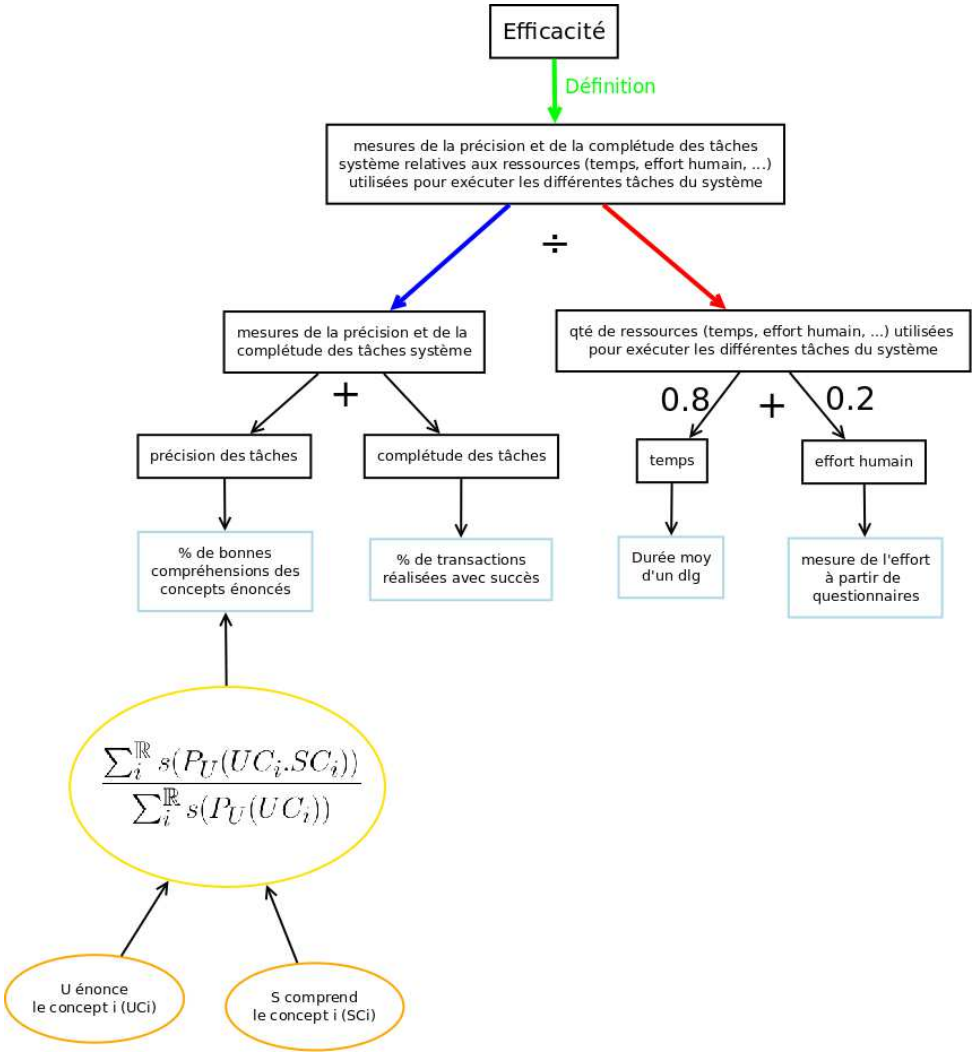


FIGURE 11 – Représentation d'un KPI mesurant l'efficacité d'un système sous forme d'arbre (les noeuds entourés d'un rectangle sont des réels tandis que ceux entourés d'une ellipse sont des éléments de $\mathcal{F}$)

À l'aide de cette représentation en arbre ou en formule, il est relativement aisé de

détecter les différences dans les constructions de KPI lorsque deux évaluateurs se basent sur un même objectif. La figure 12 page 28 représente deux KPI issus d'un même objectif : « le système est-il efficace ? ». Où se situent les différences entre ces deux KPI ? Nous pouvons les repérer en lisant les arbres depuis la racine. La première différence se situe après la division. En effet l'évaluation 2 tient compte de la complétude des tâches contrairement à l'évaluation 1 qui ne regarde que la précision des tâches. De même pour calculer les ressources utilisées, l'évaluation 2 tient compte de l'effort humain en plus du temps du dialogue. Si l'on descend encore dans l'arbre, on aperçoit d'autres différences plus fines, notamment dans le calcul de la précision des tâches et du temps du dialogue. Ainsi en continuant à descendre on peut continuer à comparer de plus en plus finement les deux évaluation.

Cependant à ce stade nous pouvons déjà faire quelques remarques critiques. En effet la représentation des évaluations est relativement claire et explique en langage naturel (du moins au début) chaque niveau de l'arbre. Ainsi la comparaison en est simplifiée, mais l'on peut supposer que ces justifications à chaque niveau ne seront pas toujours disponibles, peuvent être ambigues, et ne sont pas automatisables. Afin d'enlever les ambiguités et rendre la comparaison automatisable, nous ne pouvons nous intéresser qu'à la structure de l'arbre en recherchant des sous-arbres identiques par exemple.

Ce type de comparaison est sévère. En effet elle est binaire et ne permet de repérer que les motifs ou sous-arbres identiques. Mais deux sous-arbres peuvent être très proches sans pourtant être identiques. Prenons l'exemple du calcul des ressources utilisées. Nous avons vu que l'évaluation 2 tient compte d'un critère supplémentaire par rapport à l'évaluation 1, qui est l'effort humain. Une comparaison comme décrite précedemment nous indique seulement que les deux approches sont différentes mais nous aimerions pouvoir dire si les deux apportent des résultats complètement différents ou bien s'ils restent proches malgré tout. Pour cela nous allons nous intéresser à un calcul de similitude entre deux noeuds a priori différents en nous restreignant à l'arbre dans $\mathcal{F}$.

Afin d'aborder au mieux la recherche de mesures de comparaison de vecteurs, introduisons la notion de hiérarchie qui nous sera utile dans la suite.

5.2 Définition d'une hiérarchie

Une hiérarchie est définie par la donnée d'une suite finie de vecteurs de $\mathcal{F}$ d'ordre décroissant. Chaque vecteur a un sous-espace vectoriel de $\mathcal{F}$ associé. Ainsi les sous-espaces vectoriels associés sont de dimensions décroissantes. Tout vecteur de $\mathcal{F}$ est alors projetable dans chacun de ces sous-espaces.

Un exemple d'une hiérarchie $\mathcal{H}_0$, que nous utiliserons dans la suite est :

1. « est un dialogue »
2. « est une phase du dialogue »
3. « est un échange »
4. « est un tour de parole »
5. « est un concept »
6. « est une phrase »

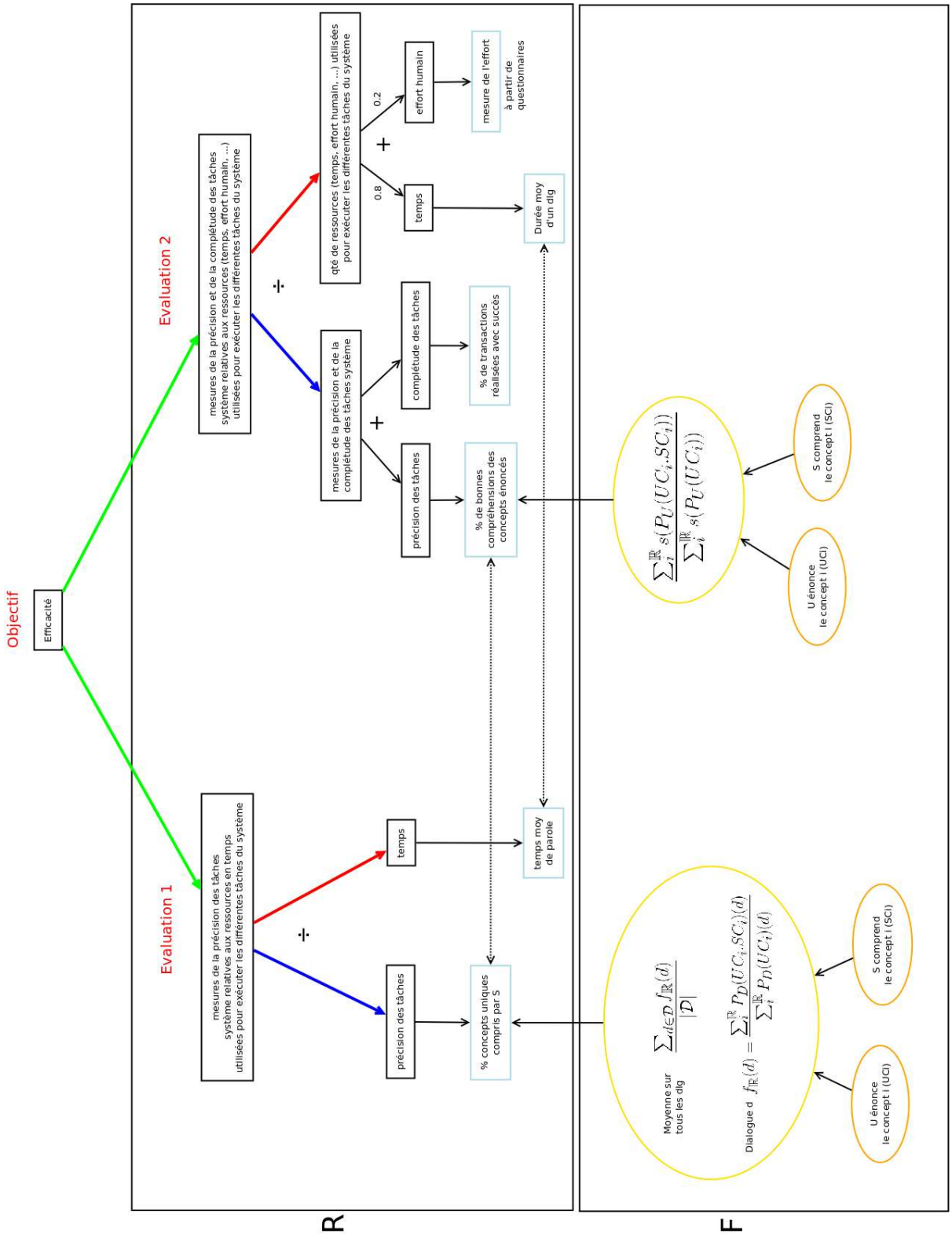


FIGURE 12 – Représentation de l'arbre de construction de deux KPI – Les constructions dans $\mathcal{F}$ et dans $\mathbb{R}$ sont mises en évidence

7. « est un mot »
8. « est une milliseconde »

phase du dialogue : découpage du dialogue en phases telles que la phase d'accueil de l'utilisateur par le système, phase d'énoncé du motif d'appel, phase de clôture,...

échange : paire de tours de parole successifs en rapport l'un avec l'autre pris par chaque participant au dialogue [3].

tour de parole : séquence de parole prononcée par un participant à un dialogue, entre le moment où ce participant commence à parler et le moment où un autre participant prend la parole [1].

concept : unité sémantique relative à l'accomplissement de la tâche. Ex : nom d'une ville, heure de départ, etc. [2]

En explicitant les termes moins consensuels, nous pouvons nous convaincre qu'il s'agit bien d'une hiérarchie. En effet, un dialogue est constitué d'une ou plusieurs phases, qui sont faites de plusieurs échanges, etc... l'ordre des différents vecteurs est donc décroissant. Un vecteur $f \in \mathcal{F}$, représentant « l'utilisateur pose une question » par exemple, est alors projetable sur chacun des espaces vectoriels associés aux différents vecteurs de la hiérarchie. Au niveau huit, $f(t)$ vaudra 1 lorsque la milliseconde t fait partie d'une question. Au niveau 6, $f(t)$ vaudra 1 lorsque la phrase t est une question et au niveau 1 $f(t)$ vaudra 1 lorsque le dialogue t contient au moins une question. Le « au moins une question » est dû au produit scalaire utilisé qui est $\forall a, b \in \mathcal{F},\ \langle a, b\rangle = \int^{+}(a \cdot b)$. Voyons maintenant comment cette représentation nous permet de comparer efficacement deux vecteurs.

5.3 Mesures de similitude entre vecteurs

Si l'on souhaite comparer deux KPI, nous ne pouvons pour l'instant que nous limiter à une comparaison simple où tous les vecteurs de bases ont le même poids, dans la mesure où lorsqu'on compare deux KPI, leurs vecteurs de base sont soit communs soit non même si un vecteur aura plus d'influence sur le résultat final qu'un autre. Cette vision binaire nous empêche d'avoir une notion subtile de la similitude entre vecteurs. Alors que deux vecteurs a priori différents dans le modèle peuvent sémantiquement correspondre à des concepts très proches. Essayons de trouver une mesure de la similitude formelle de deux vecteurs de $\mathcal{F}$ qui soit la plus cohérente possible avec la similitude au niveau du sens qu'elles portent. Nous allons pour cela émettre les hypothèses suivantes :

1. quel que soit le sous-espace vectoriel de projection, deux vecteurs égaux formellement sont proches sémantiquement
2. deux vecteurs égaux dans un sous-espace $\mathcal{G}$ de $\mathcal{F}$ sont d'autant plus proches sémantiquement que la dimension de $\mathcal{G}$ est grande
3. $\forall x \in [\![0; n]\!],\ \forall f \in \mathcal{F},\ f_x$ est d'autant plus semblable à f que x est petit

La première hypothèse exprime le fait que deux propriétés qui surviennent toujours simultanément sont très liées. La deuxième indique que deux propriétés qui surviennent simultanément à la milliseconde près sont plus liées que deux propriétés qui surviennent simultanément à la minute près. Et enfin la troisième hypothèse exprime le fait que deux propriétés « décalées dans le temps » peuvent être proches si le décalage est petit.

Considérons maintenant deux vecteurs $f, g \in \mathcal{F}$ ainsi que leur représentation dans la hiérarchie $\mathcal{H}_0$. C'est-à-dire que l'on s'intéresse à leur projection dans chacun des sous-espaces vectoriels de $\mathcal{H}_0$. Nous pouvons définir deux sortes de mesures de similitude entre f et g en accord avec les hypothèses précédentes. Une première intra-niveau hiérarchique et une deuxième inter-niveaux hiérarchiques. En effet nous pouvons définir une mesure à l'intérieur d'un niveau, c'est-à-dire en ne considérant qu'un seul sous-espace dans lequel représenter f et g, ou au contraire en tenant compte de tous les sous-espaces de la hiérarchie.

Dans le cas des mesures intra-niveau nous pouvons introduire :

$$\forall f, g \in \mathcal{G}\ s_1(f, g) = \mathcal{N}(f \cdot g)$$

ainsi que $\forall f, g \in \mathcal{G}$

$$\begin{aligned} c_1 &= min(argmax_{i \in [\![0;n]\!]}\ \mathcal{N}(\overline{f \oplus g_i})) \\ c_2 &= min(argmax_{i \in [\![0;n]\!]}\ \mathcal{N}(\overline{f_i \oplus g})) \\ s_2(f, g) &= max(c_1, c_2) \end{aligned}$$

s_1 compte le nombre d'échantillons temporels où f et g valent tous deux 1 ou tous deux 0 et s_2 fait la même chose avec un peu plus de flexibilité. On peut aisément donner d'autres mesures plausibles aussi comme $s_1(f, g) = \mathcal{N}(\overline{f \oplus g})$ ou $\frac{2 \cdot \mathcal{N}(f \cdot g)}{\mathcal{N}(f) + \mathcal{N}(g)}$ (à valeurs dans $[0; 1]$ qui vaut 1 si $f = g$ et 0 si f et g sont disjoints) mais il est difficile de juger sur la plus adaptée a priori. Notons que l'opérateur « $\cdot$ » ne tient compte que les « 1 » en commun alors que l'opérateur « $\overline{f \oplus g}$ » tient compte également des « 0 ». Ces deux mesures rendent compte des hypothèses 1 et 2 et s_2 rend compte également de l'hypothèse 3 car s_2 tient compte du décalage dans le temps grâce à g_i et f_i. Cependant ces trois hypothèses sont strictes et inflexibles sur la notion d'égalité. En effet les deux premières ne tiennent compte que d'évènements qui se produisent exactement au même moment. La troisième offre un peu plus de liberté en tenant comtpe des évènements décalés dans le temps mais impose que ce décalage soit exactement le même à chaque fois que ces évènements se produisent et introduit une rigidité à ce niveau-là. Nous aimerions introduire une mesure où deux propriétés sont liées si elles se produisent « à peu près » au même moment, où cette notion « d'à peu près » soit flexible mais très bien définie. Nous allons pour cela introduire les mesures inter-niveaux.

Dans le cas des mesures inter-niveaux, nous tiendrons compte des similitudes à tous les niveaux hiérarchiques. En effet, certains évènements n'ont de l'influence sur d'autres qu'à des niveaux hiérarchiques définis. Prenons un exemple de dialogue simple où l'utilisateur cherche à connaître les horaires de bus passant à l'arrêt La Pérouse reporté sur la figure 13.

S1 : Bonjour, je suis le système d'informations des horaires de bus de Brest.
U1 : Bonjour, je souhaite connaître l'horaire de passage du prochain bus.
S2 : Pouvez-vous préciser l'arrêt ?
U2 : À l'arrêt La Pérouse.
S3 : Le prochain bus passera à 17h52 à l'arrêt La Pérouse.
U3 : Merci, au revoir.

FIGURE 13 – Exemple de dialogue entre un utilisateur et un SDS donnant les informations sur les horaires de bus à Brest

La réplique U1 aura une influence très forte évidemment sur la réponse S2 du système. Mais elle aura également une influence (même si on peut supposer de plus en plus faible) sur les répliques qui suivent. Formellement, en appelant $u1$ et $s2$ les vecteurs « l'utilisateur énonce le concept 'demande heure de passage du prochain bus' » et « le système énonce le concept 'demande lieu de départ' », en considérant le niveau « milliseconde » et la projection s_1, $s_1(u1, s2) = 0$ et donc la liaison entre U1 et U2 n'est pas exprimée. Mais si l'on monte dans la hiérarchie au niveau « échange » par exemple, $s_1(u1, s2) = 1$. On peut alors proposer une mesure de similitude plus subtile tenant compte de tous les niveaux d'une hiérarchie $\mathcal{H}$ à k niveaux définissant k espaces vectoriels de dimension D_i et en introduisant une pondération $(w_i)_{i\in[\![1;k]\!]}$ sur chaque niveau. Ainsi $\forall f, g \in \mathcal{G}$:

$$s_{\mathcal{H}}(f, g) = \sum_{i\in[\![1;k]\!]} \frac{w_i}{D_i} \cdot s_1(f, g)$$

La notion « d'à peu près » est ici exprimée à travers les niveaux hiérarchiques, où chaque niveau hiérarchique exprime une notion d'égalité de plus en plus stricte au fur et à mesure que l'on descend dans les niveaux. Il s'agit d'une somme pondérée de la similitude intra-niveau entre deux vecteurs plongés dans chaque sous-espace vectoriel d'une hiérarchie.

En résumé nous avons construit un système descriptif des KPI, qui fournit un espace de représentation commun pour tous les KPI. Et nous avons fournit également des outils de comparaison des KPI à différents niveaux de précision. Cependant tenons toujours à l'esprit que le sens porté par un vecteur ne peut être traité que « à la main » et les mesures formelles que nous venont de voir, si elles s'efforcent d'adhérer au sens, ne tiennent comptent dans les faits que de la structure du vecteur et de l'espace dans lequel il est représenté.

6 Caractérisation des CdP

Dans cette nouvelle partie, nous n'allons plus nous intéresser aux KPI mais aux Communauté de Pratique (CdP) qui utilisent ces KPI. Nous souhaitons mieux comprendre ce qui les caractérise et les distingue afin de mieux comprendre leurs choix dans l'évaluation des SDS. En effet dans le processus (en « V ») de l'évaluation d'un SDS que nous avons vu au début de ce rapport, chaque étape de traduction d'un objectif en qualités que le système doit avoir puis en KPI est caractérisée par la CdP à laquelle appartient l'évaluateur. Celle-ci va en partie orienter certains choix et déterminer les caractéristiques du système qui vont l'intéresser.

Nous avons jusqu'à présent suivi une démarche allant du système formel vers la construction de KPI. Celle-ci nous a permis de comparer finement les différences entre KPI, mais avec cette représentation, ce que l'on gagne en précision, nous le perdons en synthèse et vision globale. En effet, pour comparer deux KPI, nous devons nous appuyer sur un arbre de construction qui peut être parfois très grand et la comparaison avec d'autres KPI n'en est que plus difficile. De plus, les différentes CdP ne se définissent pas à l'aide d'un simple KPI mais grâce à un ensemble de KPI.

Afin de comprendre les CdP et les différences entre KPI de manière plus synthétique et aisée, nous avons adopté une démarche inverse en partant des CdP existentes chez Orange Labs et en recherchant des éléments qui puissent les différencier et les caractériser à la lumière du système formel de représentation des KPI.

6.1 Étude syntaxique des KPI utilisés chez Orange Labs

Nous avions pour cela à notre disposition une liste de KPI définis en langage naturel parfois de façon peu précise, associés aux CdP qui les utilisent. Les CdP en question sont :

- **les maîtres d'ouvrage**
- **les techniciens** qui s'occupent à la fois de la conception, du développement et de l'exploitation
- **les experts de la parole** qui s'intéressent davantage à la reconnaissance vocale et l'analyse sémantique
- **les ergonomes**
- **les personnes du marketing**
- **les personnes « du métier »** : il s'agit des personnes travaillant dans le domaine où le SDS sera employé.

Les différentes CdP associées aux KPI qu'elles utilisent ont été collectées par Diane Cros pour une étude commanditée par le laboratoire NADIA d'Orange, en s'informant auprès des différentes parties prenantes aux projets SDS chez Orange Labs. Cette liste de KPI/CdP est déjà une représentation des CdP : les CdP sont constituées d'un ensemble de KPI. Cette représentation nous est imposée par le matériel dont nous disposons, est évidemment incomplète et ne rend pas compte de toutes les caractéristiques d'une CdP mais nous pouvons supposer sans grands risques certaines de ses propriétés :

- deux CdP différentes ont une représentation différente (discernabilité)
- toutes les CdP sont représentables comme un ensemble de KPI (extensivité)

En analysant le corpus entier de KPI, nous avons mis en évidence une structure syntaxique des KPI d'Orange Labs qui forment ainsi une classe de KPI plus simples que nous pourrons analyser plus facilement par la suite.

Cette classe définit des KPI constitués d'un sous-espace vectoriel de référence, d'un sous-espace vectoriel de projection et d'un filtre conditionnel qui détermine un vecteur de $\mathcal{F}$ comme illustré par l'exemple de la figure 14.

Son arbre de construction est presque entièrement dans $\mathcal{F}$. Dans $\mathbb{R}$ il s'agit d'un ratio : seul l'opérateur de division est utilisé. En détaillant sa construction, le KPI de la figure 14 est le ratio de la norme du vecteur « est un mot bien reconnu » (qui peut éventuellement être construit à partir d'autres vecteurs de $\mathcal{F}$) projeté dans le sous-espace vectoriel associé

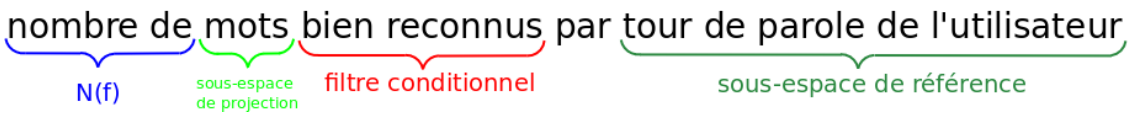


FIGURE 14 – analyse syntaxique d'un KPI

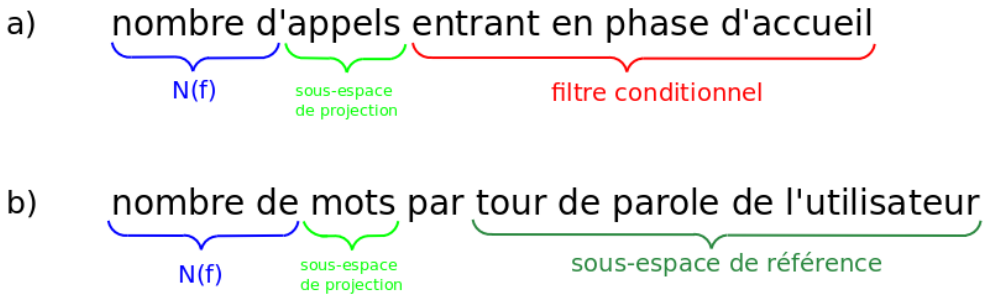


FIGURE 15 – analyse syntaxique d'un KPI

au vecteur « est un mot »par la dimension du sous-espace vectoriel associé au vecteur « est un tour de parole de l'utilisateur ».

On remarque que parfois le sous-espace de référence est omis comme dans le a) de la figure 15. Dans ce cas il est sous-entendu, on considère tout le corpus de dialogues et la dimension du sous-espace de référence vaut 1. Dans le b) de la figure 15, filtre conditionnel et sous-espace de projection sont confondus car le sous-espace de projection est issu du vecteur filtre conditionnel « est un mot ».

Ainsi nous sommes capables de représenter une CdP de manière relativement concise. Une CdP est représentée par un ensemble de KPI appartenant à la classe que nous venons de décrire et donc eux-mêmes représentables par un sous-espace vectoriel de référence, un sous-espace vectoriel de projection et un filtre conditionnel.

6.2 Caractérisation ensembliste des CdP

Mais que peut-on dire d'un nouveau KPI qui n'est encore utilisé par aucune CdP ? À quelles CdP sera-t-il utile ? Ou inversement, si une CdP recherche de nouvelles mesures, quel KPI correspondra a priori le mieux aux besoins de la CdP ? Bien sûr il n'est pas possible de répondre à ces questions de façon sûre et fiable car le contexte d'évaluation, les objectifs fixés sont autant de paramètres dont on ne peux pas tenir compte mais qui ont une grande influence sur l'évaluation.

Cependant nous pouvons tout de même trouver une caractérisation, une signature d'une CdP qui puisse nous donner des indices sur les types de KPI qu'elle utilise le plus souvent et sur ce qui la distingue le mieux des autres CdP. Nous avons alors tenté d'analyser statistiquement les CdP à partir de la liste de KPI que nous avions à disposition.

Dans un premier temps nous avons considéré les CdP d'Orange simplement comme un ensemble de KPI et nous avons observé les relations entre ces ensembles. Cette première approche ne nous a apporté que peu de choses :

– les techniciens et les experts de la parole utilisent presque toujours les mêmes KPI et mis à part 2 KPI, les KPI utilisés par les experts de la parole sont aussi utilisés par les techniciens.
– les ergonomes utilisent beaucoup de KPI communs avec les techniciens et les experts de la parole tout en gardant des KPI qui leurs sont spécifiques.
– les maîtres d'ouvrage et les personnes du métier et du marketing utilisent à la fois des KPI spécifiques à leur domaine et des KPI utilisés par les techniciens, ergonomes et experts de la parole.

Cette analyse reste assez pauvre notamment car il est impossible d'étudier efficacement des objets aussi complexes en s'en tenant à des représentations si simples. D'un autre côté la surabondance de précisions et d'information est tout autant inexploitable. C'est pourquoi nous avons tenté une deuxième approche assez simple pour pouvoir être étudiée et assez élaborée pour être intéressante, basée sur les réflexions suivantes. Les maîtres d'ouvrage et les personnes du marketing s'intéressent à des KPI « haut niveau » qui prennent en compte la globalité du SDS. À l'opposé, les techniciens et les experts de la parole vont se focaliser sur des aspects plus spécifiques qui concernent des modules ou des aspects particuliers du système. Il y aurait une notion de hiérarchie (semblable à celle que nous avons abordé précédemment) dans le choix des KPI par les CdP.

6.3 Niveau hiérarchique associé à un KPI

L'idée est alors de définir un KPI « haut niveau » ou au contraire « spécifique », une hiérarchie de niveaux et réussir à caractériser une CdP par ses préférences en termes de niveaux. Nous allons pour cela utiliser la hiérarchie $\mathcal{H}_0$. En effet des KPI « haut niveau » sont des KPI qui concernent la globalité du système, par exemple un KPI qui note l'efficacité du système, ou un KPI qui quantifie le degré de satisfaction des utilisateurs. Ils vont considérer un dialogue dans sa globalité. Beaucoup plus bas dans la hiérarchie, un KPI qui calcule le taux d'erreurs de reconnaissance va considérer non pas un dialogue dans sa globalité mais chaque mot.

Commençons par définir plus précisément le niveau hiérarchique associé à un KPI simple comme ceux que nous avons vus dans la section sur l'étude syntaxique des KPI d'Orange. Prenons l'exemple de la figure 15 a) page 33 : « nombre d'appels entrant en phase d'accueil ». Ici on distingue deux niveaux hiérarchiques : « est un appel », lié au sous-espace de projection et « est une phase », lié à la condition sur les appels, au filtre conditionnel. Lequel correspond le mieux à la notion de hiérarchie dont nous avions eu l'intuition ? Nous ne pouvons a priori pas trancher, bien que le bon sens nous conduit à préférer le niveau lié au filtre conditionnel. En effet bien que l'on compte les appels, l'information importante vient du niveau de précision de la condition sur les appels : « entrant en phase d'accueil ». De plus nous pouvons remarquer que le niveau hiérarchique associé au sous-espace de projection sera toujours supérieur au niveau lié au filtre conditionnel car des KPI tels que « nombre de phases telles que l'appel est [filtre conditionnel] »n'ont pas de sens. Les deux niveaux sont donc liés de toute façon. Nous allons donc dans la suite considérer que le niveau hiérarchique associé à un KPI simple est le niveau hiérarchique lié au filtre conditionnel.

Si l'on considère un KPI quelconque et son arbre de construction dans $\mathcal{F}$ et $\mathbb{R}$, les KPI simples sont ceux qui se trouvent à la frontière entre $\mathcal{F}$ et $\mathbb{R}$ (du côté de $\mathbb{R}$). Ces KPI simples sont ensuite utilisés pour construire le KPI global dans $\mathbb{R}$. Mais à quel niveau hiérarchique correspond le KPI dans sa globalité ? On pourrait inventer des milliers de façons différentes de calculer le niveau hiérarchique d'un KPI à partir des KPI simples qui le constituent. En voici une : soit K un KPI et N un noeud de son arbre de construction dans $\mathbb{R}$. Notons $(f_i)_{i\in[\![1,n]\!]}$ les n fils de N et h la fonction qui à tout noeud de l'arbre associe son niveau hiérarchique. Alors

$$h(N) = max_{i\in[\![1;n]\!]}(h(f_i))$$

C'est-à-dire que le noeud père a le même niveau hiérarchique que le fils ayant le plus haut niveau hiérarchique. En associant un niveau hiérarchique à chaque KPI, nous avons simplifié leur représentation assez pour pouvoir l'exploiter efficacement dans l'étude des CdP que nous allons abordé dans la section suivante.

6.4 Signature d'une CdP

Notre but depuis le début de la section était de trouver une meilleure façon de caractériser une CdP que la caractérisation ensembliste que nous avons vue et de pouvoir ainsi mieux les comprendre et les comparer. Grâce aux niveaux hiérarchiques associés aux KPI nous avons représenté les CdP comme une distribution de préférences sur les niveaux hiérarchiques. Autrement dit pour tout niveau hiérarchique h et toute CdP c nous avons calculé à partir de notre liste de KPI $P(h|c)$. Une CdP est alors caractérisée par les probabilités associées à tous les niveaux hiérarchiques. Les résultats sont reportés sur le tableau 4.

	dialogue	phase	tour de parole	concept	mot	milliseconde
maîtres d'ouvrage (MOA)	0,55	0,15	0,18	0,09	0	0,03
personnes du métier	0,52	0,05	0,26	0,11	0	0,02
techniciens (Tech)	0,30	0,24	0,12	0,16	0,15	0
ergonomes (Ergo)	0,36	0,31	0,16	0,09	0,04	0
experts de la parole (SLU)	0,22	0,20	0,18	0,16	0,22	0
marketing (Mark)	0,86	0,14	0	0	0	0

TABLE 4 – Distribution des niveaux hiérarchiques au sein des CdP d'Orange Labs

Ce tableau permet donc d'avoir un outil de comparaison et de caractérisation des différentes CdP. Nous conservons toujours la propriété d'extensivité lorsque nous représentons une CdP de cette façon mais nous perdons d'assurance sur la discernabilité. En effet si pour les CdP dont nous disposons la discernabilité est conservée, rien ne nous garantit cette propriété pour toute CdP. Cependant cette représentation est intéressante dans la mesure où elle rend bien compte des réflexions précédentes sur les CdP et ce de façon beaucoup plus claire et concise que si l'on considérait une CdP comme un ensemble de KPI.

6.5 Distance entre un KPI et une CdP

Reprenons les questions que nous nous posions au début de la section. Que peut-on dire d'un nouveau KPI qui n'est encore utilisé par aucune CdP ? À quelles CdP sera-t-il utile ? Ou inversement, si une CdP recherche de nouvelles mesures, quel KPI correspondra a priori le mieux aux besoins de la CdP ? Dans cette nouvelle partie nous allons un peu plus loin en essayant d'intégrer les changements potentiels que peut connaître une CdP dans le temps. En effet une CdP ne peut être représentée de façon figée par un ensemble de KPI. Cet ensemble évolue sans cesse en intégrant de nouveaux KPI et en en délaissant d'autres.

Grâce à la signature d'une CdP que nous avons exhibée précédemment à travers les distributions de préférences, nous pouvons tenter de répondre à ces questions en cherchant une mesure a, d'acceptation a priori d'un KPI K inclassé (représenté par son arbre dans $\mathbb{R}$) par une CdP C (représentée par $(c_i)_{i\in[\![1,m]\!]}$, les probabilités associées à chaque niveau hiérarchique d'une hiérarchie H à m niveaux).

Soit N un noeud de l'arbre de construction dans $\mathbb{R}$ de K. Notons $(f_i)_{i\in[\![1,n]\!]}$ les n fils de N et h la fonction qui à tout noeud de l'arbre associe son niveau hiérarchique. On peut également étiqueter l'arbre par des poids notons w_i le poids lié au fils f_i (entre 0 et 1, typiquement, $\forall i \in [\![1;n]\!]\ w_i = \frac{1}{n}$). Alors

$$a(C,N) = c_{h(N)} + \sum_{i=1}^{n} w_i \cdot a(C,f_i)$$

Il s'agit d'une définition récursive partant de la racine, où chaque noeud de l'arbre a d'autant plus d'influence sur le résultat final que sa profondeur est petite et que sa pondération est grande (c'est-à-dire en considérant la pondération $\forall i \in [\![1;n]\!]\ w_i = \frac{1}{n}$, que le nombre de noeuds voisins de même profondeur est petit). Le noeud racine sera donc celui qui prédominera.

Dans l'exemple de la figure 16 page 37, nous avons représenté au centre l'arbre de construction dans $\mathbb{R}$ d'un KPI calculant l'efficacité. Nous avons ensuite étiqueté chaque noeud de cet arbre par son niveau hiérarchique comme nous l'avons vu dans la section 6.3. Cet étiquetage est représenté par les couleurs de remplissage des noeuds : les niveaux hiérarchiques sont le dialogue, la phase et le concept. Puis nous avons calculé l'adaptation de ce KPI à chaque CdP (représentée par un rectangle coloré qui illustre sa distribution de préférences) ce qui donne les résultats en rouge pour chaque CdP. On remarque que sémantiquement les résultats sont cohérents. En effet l'efficacité d'un système est une mesure qui intéresse avant les personnes du marketing et les maîtres d'ouvrage que les experts de la parole par exemple. Nous retrouvons cette idée à travers les résultats.

De cette manière, nous avons introduit une façon de modéliser la variabilité des CdP dans le temps grâce à cette mesure qui permet d'inclure de nouveaux KPI dans une CdP. Cependant, remarquons que cette méthode est évidemment trop simple pour être fiable et ne peut être justifiée que par son application concrète sur des cas réels.

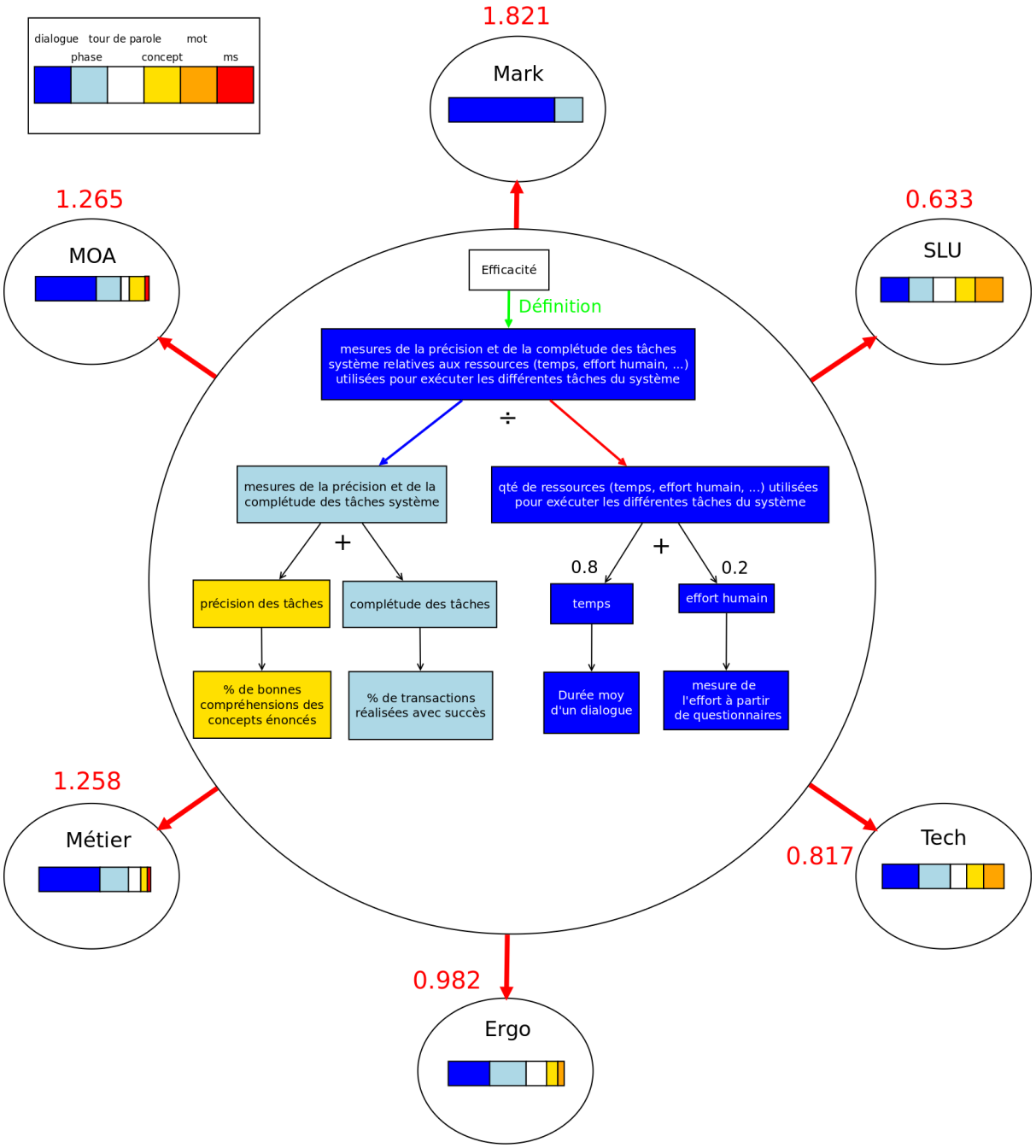


FIGURE 16 – Calcul de l'acceptation d'un nouveau KPI calculant l'efficacité, parmi les CdP d'Orange Labs

Conclusion

Nous avons introduit un système formel de représentation, de description de certains aspects du processus d'évaluation, notamment des KPI et des CdP. Placés dans ce système, ces objets nous ont montrés certaines caractéristiques intéressantes qui nous ont permi de mieux comprendre leur sens, de les comparer et de repérer leurs différences sans pour autant les juger et indiquer lequel serait « meilleur » qu'un autre. Mais nous avons

également discerné certaines limites dues à cette représentation qui a déformé et supprimé une partie de l'information qu'ils contenaient car nous n'avons représenté qu'une partie de leur sens. Au sein du processus d'évaluation nous avons tenté de modéliser un autre élément qui sont les CdP à travers leurs préférences en termes de KPI ainsi que leur comportement vis-à-vis de KPI connus ou inconnus.

Après ces études ce qui manque est un moyen de justifier et tester ce système dans un contexte réel car les données dont nous disposions étaient issues principalement de la litérature. Ceci permettrait de l'améliorer en pointant certaines failles ou incohérences par rapport à une situation réelle.

De plus ces éléments ainsi que les cas de figure que nous avons traités ne constituent pas à eux seuls toute l'évaluation d'un SDS. D'autres éléments connus comme l'objectif d'évaluation ou inconnus, irrationnels ou aléatoires dûs au contexte complexe entourant l'évaluation jouent un rôle majeur et ne peuvent être étudiés efficacement que très difficilement.

Annexes

Abbr.	Name	Definition	Int. level	Meas. meth.
TS	task success	Label of task success according to whether the user has reached his/her goal by the end of a dialogue, provided that this goal could be reached with the help of the system. The labels indicate whether the goal was reached or not, and the assumed source of problems: • *TS:S*: Succeeded (task for which solutions exist) • *TS:SCs*: Succeeded with constraint relaxation by the system • *TS:SCu*: Succeeded with constraint relaxation by the user • *TS:SCsCu*: Succeeded with constraint relaxation both from the system and from the user • *TS:SN*: Succeeded in spotting that no solution exists • *TS:Fs*: Failed because of the system's behaviour, due to system in adequacies • *TS:Fu*: Failed because of the user's behaviour, due to non-cooperative user behaviour See also [8][7][24].	dial.	expert.
κ	kappa coefficient	Percentage of task completion according to the kappa statistics. Determined on the basis of the correctness of the result AVM reached at the end of a dialogue with respect to the scenario (key) AVM. A confusion matrix *M(i,j)* is set up for the attributes in the result and in the key, with *T* the number of counts in *M*, and t_i the sum of counts in column *i* of *M*. Then $$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$ with *P(A)* the proportion of times that the AVM of the actual dialogue and the key agree, $P(A) = \sum_{i=1}^{n} \frac{M(i,i)}{T}$. *P(E)* can be estimated from the proportion of times that they are expected to agree by chance, $P(E) = \sum_{i=1}^{n} (\frac{t_i}{T})^2$. [31][4]	dial. or set of dial.	expert.

FIGURE 17 – Paramètres liés aux tâches

Références

[1] N. O. Bernsen, H. Dybkjaer, and L. Dybkjaer. *Designing Interactive Speech Systems : From First Ideas to User Testing*. Springer, 1998.

[2] Boros et al. Towards understanding spontaneous speech word accuracy vs. concept accuracy. 1996.

[3] N Fraser. Assessment of Interactive Systems. In D Gibbon, R Moore, and R Winski, editors, *Handbook on Standards and Resources for Spoken Language Systems*, pages 564–615. Mounton de Gruyter, Berlin, 1997.

[4] H.P. Grice. Logic and conversation. In *Syntax and semantics*, volume 3, pages 41–58. New York : Academic Press, 1975.

[5] K.S. Hone and R. Graham. Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, 6 :287–303, 2000.

[6] ITU-T Rec. P.Sup24. Parameters describing the interaction with spoken dialogue systems, 2005.

[7] ITU-T Rec.P.851. Évaluation subjective de la qualité des services téléphoniques basés sur des dialogueurs automatiques, 2003.

[8] M. Laurent and P. Bretier. MPOWERS : a Multi Points Of VieW Evaluation Refinement Studio. *Computational Linguistics*, pages 265–268, 2010.

[9] M. Laurent, P. Bretier, and I. Kanellos. Considering the subjectivity to rationalise evaluation approaches : the example of Spoken Dialogue Systems. *QoMEx10 - Second International Workshop on Quality of Multimedia Experience*, 2010.

[10] Marianne Laurent, Philippe Bretier, and Carole Manquillet. Ad-hoc evaluations along the lifecycle of industrial spoken dialogue systems : heading to harmonisation ? *LREC 2010 : 7th international conference on Language Resources and Evaluation*, 2010.

[11] S. Möller. *Quality of telephone-based spoken dialogue systems*. Springer Verlag, 2005.

[12] T. Paek. Toward evaluation that leads to best practices : reconciling dialog evaluation in research and industry. *Workshop on Bridging the Gap : Academic and Industrial Reasearch in Dialog Technologies*, pages 40–47, 2007.

[13] R. Pieraccini and J. Huerta. Where do we go from here ? Research and commercial spoken dialog systems. In *6th SIGdial Workshop on Discourse and Dialogue*. ISCA, 2005.

[14] R. Pieraccini, D. Suendermann, J. Liscombe, and K. Dayanidhi. Are We There Yet ? Research in Commercial Spoken Dialog Systems. In *Text, Speech and Dialogue, 12th International Conference*, pages 3–13, 2009.

[15] R. Poppe, R. Rienks, and B. Van Dijk. Evaluating the Future of HCI : Challenges for the Evaluation of Emerging Applications. *Computing*, pages 234–250, 2007.

[16] M.A. Walker, D.J. Litman, C.A. Kamm, and Alicia Abella. PARADISE : a framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 271–280, 1997.