

Spécifications du langage DAFT 2.0

François Bouchet – juin 2006

SYNTAXE DES REQUETES	3
CLASSES DE REQUETES	3
TYPES	4
SCHEMAS DES ACTIONS GENERIQUES	5
SCHEMAS DES FONCTIONS COMMUNICATIVES	7
HELLO	7
BYE	7
PING	8
ACK	9
SCHEMAS DES MODALITES	10
INFOS ⇔ a / o	10
MEANING ⇔ o	11
TYPE	12
VALUE ⇔ o	13
PROPERTY ⇔ o	14
FUNCTION ⇔ o	15
ROLE	16
FUNCTIONING ⇔ o	17
LIMIT	18
POSSIBILITY ⇔ o	19
KNOWLEDGE ⇔ a / o	22
OBLIGATION ⇔ a	23
WILL	24

PROBABILITY	25
EXISTENCE ⇔ o	26
OTHER ⇔ a / o	27
OBJECT ⇔ o	28
DIFFERENCE ⇔ o	29
ORDER ⇔ a	30
WAY ⇔ o	32
EFFECT ⇔ o	33
REASON ⇔ o	34
PLACE ⇔ o	35
MOMENT ⇔ o / t	36
NUMBER	37
LIKE	38
FEAR	39
BOTHER	40
DOUBT	41
SURPRISE	42
REGRET	43
HAPPY	44
PROBLEM ⇔ o	45
MISTAKE ⇔ o	46
HELP ⇔ o / a	47
TELL ⇔ a	48
CHECK ⇔ a	49
ASK	49
NEG	49
DE LA PROPAGATION DES MARQUEURS	50

Syntaxe des requêtes

Les requêtes DAFT s'écrivent sous la forme :

$M1 (\dots Mn (c1 = A1 (c'1 = xxx, \dots, c'l = yyy), \dots, cm = Am (\dots)) \dots)$

Qu'on peut écrire sous la forme $M(A())$ pour en alléger l'écriture, expression dans laquelle :

$M1 - Mn$ sont des modalités

$A1 - Am$ sont des actions

$c1 - cm$ & $c'1 - c'l$ sont les intitulés des champs tels que décrits dans les schémas des modalités et des actions

xxx & yyy sont des éléments issus de la requête textuelle

Classes de requêtes

On distingue 4 classes de requêtes faites à l'agent :

- les commandes de contrôle, correspondant à des ordres simples à exécuter.
- les demandes d'assistance directes, essentiellement sous forme de questions.
- les demandes d'assistance indirectes, qui sont des assertions sous-entendant une difficulté pour atteindre un objectif.
- les éléments liés au dialogue utilisateur-agent.

Types

Les types constituent une distinction faite entre les mots en fonction de la nature de ce à quoi ils réfèrent. Le typage des groupes lexicaux (effectué en amont dans la chaîne de traitement) permet de limiter de facto les possibilités de remplissage des champs des schémas d'action et de modalité, mais aussi leur imbrication, la plupart des schémas ayant aussi un type global auquel ils sont assimilés et qui est noté " \Leftrightarrow TYPE".

- a** action
- c** concept abstrait
- o** un objet physique de l'application
- P** propriété
- v** valeur
- T** type
- p** personne (**utilisateur** ou **système**)
- m** manière
- l** un lieu ou une zone de l'application, notamment dans le cadre de la navigation
- t** temps (*jamais* = 0, *passé* = 1, *présent* = 2, *futur* = 4, *toujours* = 7)
 - | **ti** instant
 - | **td** durée

Schémas des actions génériques

Les actions génériques, comme leur nom le laisse entendre, sont toutes de type global **a**. Elles sont au nombre de 53, dont voici les noms et schémas associés :

Actionner([o]objet)
Activer([o]objet)
Adhérer([p]pA, [o]objet)
Ajouter([o]objet, [l]lieu, [m]manière)
Aller([l]lieu, [ti]instant, [m]manière)
Annuler([a]action, [o]objet)
Arrêter([a]action, [o]objet, [ti]instant, [td]durée)
Bipper()
Bouger([o]objet, [p]personne, [l]dans, [l]origine, [l]destination, [m]manière)
Cacher([o]objet)
Changer-de([o]objet)
Cliquer([p]personne, [o]objet, [m]manière)
Compter([m]manière)
Contacter([p]personne)
Controler([o]objet, [P]propriété)
Coup([l]origine, [l]destination)
Créer([o]objet)
Démarrer([o]objet)
Dépasser([o]objet, [P]propriété, [v]valeur)
Donner([p]donneur, [p]receveur, [o]objet)
Enregistrer([o]objet, [l]lieu)
Etre-perdu([p]personne)
Etre-prêt([p]personne)
Etre-utile([o]objet)
Faire-défiler([o]objet)
Faire-partie-de([o]objet,[o]contenant)
Fermer([o]objet, [ti]instant)
Finir()
Gagner([m]manière)
Jouer([p]joueur, [p]pour, [o]objet, [ti]instant, [m]manière)
Manipuler([p]manipulateur, [o]objet, [m]manière)
Mettre-à-jour([o]objet, [p]pA)
Modifier([o]objet, [P]propriété, [v]valeur, [m]manière)
Montrer(nég, [p]pAQuiMontre, [p]pOquiEstMontré, [o]oAMontrer, [m]manière, [l]lieuOùMontrer)
Quitter([o]objet)
Recommander([o]objet)
Recommencer([c]concept)
Redémarrer([o]objet, [p]personne)
Rejouer([c]concept)
Restaurer([o]objet, [v]valeur)
Réversible([a]action, [c]concept)
Se-abonner()
Se-charger-de([p]responsable,[a]action)
Se-passer([ti]instant)
Supprimer([o]objet, [m]manière)
Télécharger([o]objet)
Tricher([m]manière)
Trier([o]objet, [m]manière)
Utiliser([o]objet)

Vérifier([o]objet)

Voir([o]objet, [p]personne)

V*()

Schémas des fonctions communicatives

Les fonctions communicatives ont un statut un peu à part par rapport aux autres : elles ne s'imbriquent pas et ne peuvent contenir d'autres schémas, par conséquent, elles n'ont pas de type global équivalent.

HELLO

Structure

Salutation de début de conversation.

HELLO()

Commentaires

-

Exemples

Bonjour
Hello
Salut

BYE

Structure

Salutation de fin de conversation.

BYE()

Commentaires

-

Exemples

Bye
ciao (bella)
au revoir
je me casse
end (svp)
à une prochaine fois
auf viedersen

PING

Structure

Vérification de présence et d'activité de son interlocuteur.

PING()

Commentaires

-

Exemples

[nom de l'agent] ?

où es-tu ?

es-tu là ?

pas vrai ?

il y a quelqu'un ?

tu dors ?

ACK

Structure

Accusé de réception et accord d'une personne au sujet de quelque chose.

[p]	personne	personne acceptante
[o]	objet	ce qui est accepté

Commentaires

On ne fait pas de distinction entre le ACK, AGREE et THANKS, qui sont souvent confondus, exprimant tous l'accord et la satisfaction de l'utilisateur, même implicite. Ce choix se justifie par l'observation que l'utilisateur ne manque pas d'exprimer sa frustration ou son désaccord si la réponse ne lui convient pas. Tout comme pour POSSIBILITY, il s'agit d'une approche très pragmatique : on perd une subtilité importante dans l'absolu, mais dont le rôle est somme toute minime dans le cadre considéré.

Exemples

Ça va ([pas trop mal/pas si mal que ça])

c'est (pas) ça

OK (pour moi)

Merci

Parfait !

Ca marche

Bon / Bien / Absolument / Bien sûr / Certainement / Sûrement / Super / Très juste

ne pas faire dire

excellente idée

approuver

convenir (tout à fait)

pas trop mal

tu peux le dire

Non (alors)

c'est totalement faux

Pas question

Pas d'accord

J'y suis pas

Etre du même avis sur

Pas question

Je proteste

Etre [pour/contre]

Schémas des modalités

INFOS ⇔ a / o

Structure

(Avoir) des informations/**plus** d'informations sur un **sujet**.

[c]	about	concept considéré
[o]	<i>about</i>	objet considéré
bool	more	plus d'infos (0 / 1)

Commentaires

-

Exemples

(objet = objet + concept)

Qui/Quel est + objet

C'est quoi + objet

Qu'est-ce que (c'est) + objet

Que peux-tu dire au sujet de + objet

Des informations/De la documentation sur/De l'info sur/au sujet de + objet

Plus d'infos / Plus d'explications / Plus de documentation sur / Une explication supplémentaire sur + objet

Parle moi un peu de + objet

Soit plus précis sur + objet

Renseigner au sujet de + objet

MEANING ⇔ ○

Structure

Le sens **de** quelque chose, **d'après** quelqu'un.

[c]	of	concept considéré
[o]	<i>of</i>	objet considéré
[p]	according	personne définissant

Commentaires

-

Exemples

(personne) + vouloir dire + objet
Objet + avoir une signification (précise)
Etre la signification de + objet
Objet + correspondre à
Signifier + objet

TYPE

Structure

Le type **d'**un objet **est** type.

[o]	of	objet considéré
[T]	is	type affirmé ou supposé

Commentaires

-

Exemples

Objet + être + type

Qu'est-ce que (c'est) + objet

VALUE ⇔ ○

Structure

La valeur (de la **propriété** d'un **objet** / d'un objet) **est** valeur.

[o]	object	objet considéré
[P]	property	propriété considérée de l'objet
[v]	is	valeur de la propriété de l'objet
bool	multiple	valeur unique (0) ou ensemble de valeurs (1)
-	asked	valeur demandée, champ à valeurs dans : { current , min, max}

Commentaires

-

Exemples

Propriété d'un objet + être activable/visible

Le minimum/maximum de/pour + objet

La valeur de + objet/propriété d'un objet

Valeur + être le nom de + objet

Objet + s'appeler + valeur

PROPERTY ⇔ o

Structure

La propriété '**propriété**' d'un objet **vaut** une valeur.

[o]	of	objet considéré
[P]	is	propriété considérée de l'objet
[v]	worth	valeur de la propriété de l'objet
bool	multiple	propriété unique (0) ou ensemble de propriétés (1)

Commentaires

-

Exemples

Les attributs/propriétés de + objet

FUNCTION ⇔ ○

Structure

La fonction **faisant** une action déterminée **pour** des objets donnés **dans** un cadre particulier, et **ayant** certaines propriétés.

[a]	doing	action effectuée par la fonction
[o]	for	objets devant avoir cette fonction
[l]	in	lieu dans lequel doit se trouver cette fonction
[P]	being	propriétés de la fonction

Commentaires

Il peut être difficile de faire corroborer la description de la fonction avec une action connue du système.

Exemples

Une fonction/commande + action+ pour + object + dans + place

De fonctions + property

Des fonctions de + action

ROLE

Structure

Le rôle **d'**un objet / concept / personne **par rapport à** un référentiel.

[o]	of	objet considéré
[c]	<i>of</i>	concept considéré
[p]	<i>of</i>	personne considérée
[o]	regarding	objet par rapport auquel le rôle est considéré

Commentaires

-

Exemples

A quoi correspondre/servir + objet

But de + objet

Objet + agir sur + référentiel

Qu'est ce que c'est + objet

Objet + pour quoi faire

Rôle de + objet + par rapport à + référentiel

FUNCTIONING ⇔ o

Structure

Le fonctionnement d'un objet.

[o]	of	objet considéré
[c]	<i>of</i>	concept considéré

Commentaires

-

Exemples

Comment marcher/fonctionner + objet
Le fonctionnement/comportement de + objet
Ce qui se passe avec/durant + concept

LIMIT

Structure

Limites **dans** un objet / **des** propriétés d'un objet.

[o]	in	objet considéré
[c]	<i>in</i>	concept considéré
[P]	of	propriété considérée de l'objet

Commentaires

-

Exemples

Objet + être limité

Y avoir une limite / des limitations (à / de) + objet

POSSIBILITY ⇔ o

Structure

Possibilités offertes à quelqu'un **par** une autorité (**de** réaliser quelque chose / **au sujet d'**un objet) **pour** quelqu'un.

[p]	of	personne dont on considère les possibilités
[p]	for	personne pour qui on considère les possibilités
[p]	offered-by	autorité offrant les possibilités
[a]	todo	action considérée
[o]	about	objet considéré
[t]	time	temps considéré

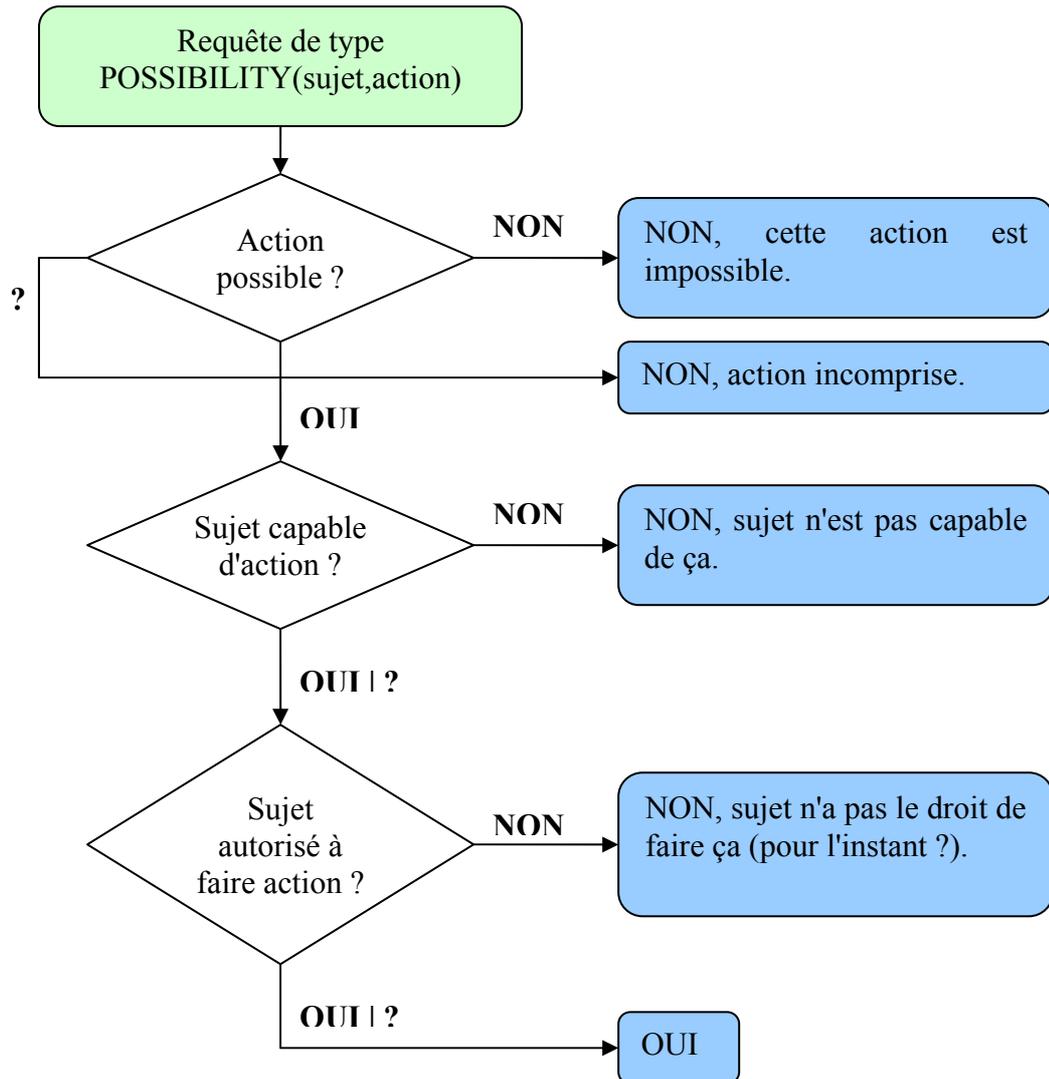
Commentaires

Il existe une forte ambiguïté difficile à lever sur "pouvoir", qui peut avoir valeur de possibilité (épistémique) ou de probabilité (déontique).

En théorie, capacité, permission et possibilité sont 3 significations indépendantes. Néanmoins, dans le cadre de l'assistance, lorsque l'on demande "peux-tu modifier le tableau ?", on veut savoir à la fois :

1. Si c'est une action est possible.
2. Si le sujet a les capacités et les connaissances nécessaires pour réaliser l'action.
3. Si le sujet est autorisé à faire l'action à l'instant de la demande.

Selon les maximes de quantité de Grice, la réponse de l'agent se doit d'être "aussi informative que requis par le contexte", or la probabilité est forte qu'en cas de réponse affirmative à la question ci-dessus, l'utilisateur demande immédiatement après de la réaliser, et une réponse de type "Je ne sais pas le faire" ou "Je ne suis pas autorisé à faire ça pour l'instant" serait très certainement frustrante pour l'utilisateur, d'où la nécessité d'analyser les requêtes de ce type selon ces 3 significations.



Ce sera le rôle du module pragmatique que d'analyser ces 3 composantes de la possibilité, qui ne sont bien sûr pas toujours toutes pertinentes.

Exemple 1 : "puis-je modifier le tableau ?"

1. Le système vérifie que c'est une action connue comme possible.
2. Le système ne peut pas savoir si l'utilisateur a la capacité de faire l'action : il suppose que oui.
3. Le système vérifie que c'est une action autorisée pour l'utilisateur à l'instant de la demande.

Exemple 2 : "peux-tu modifier le tableau ?"

1. Le système vérifie que c'est une action connue comme possible.
2. Le système vérifie que l'agent a la capacité de faire l'action.
3. Le système vérifie que l'agent est autorisé à accomplir l'action à l'instant de la demande.

Exemple 3 : "peut-on modifier le tableau ?"

1. Le système vérifie que c'est une action connue comme possible.
2. Le système vérifie les personnes ayant la capacité de faire l'action.
3. Le système vérifie que ces personnes soient autorisées à faire l'action.

Exemples

que pouvoir (je/tu/on) pour
être possible/impossible/dangereux/permis/autorisé/capable/à même de
ce que je peux faire
ce qui est proposé
que proposer
quoi faire
rien à faire
choix possibles

KNOWLEDGE ⇔ a / o

Structure

(Avoir) La connaissance / Le savoir / La compréhension / La capacité **de** quelqu'un ou quelque chose (**pour** faire une action / **au sujet d'**un objet), avec un certain **degré** de certitude.

[p]	of	personne ayant le savoir / la connaissance
[o]	of	objet ayant la capacité
[a]	todo	action dont on considère la connaissance
[o]	about	objet considéré – mot clé "all"
[t]	time	temps considéré
bool	certainty	degré de certitude sur le savoir (0=croyance, 1= certitude)

Commentaires

Ambiguïté sur l'équivalence du type, qui peut être action ou objet selon les cas.

Exemples

Saisir

Connaître

Arriver à

Savoir

Comprendre

Avoir un tuyau pour

Avoir une solution

Avoir connaissance de

Se figurer + objet

Etre sûr de (certainty = 1)

Ne pas être sûr de (certainty = 0)

Je pense / Il me semble que (certainty = 0)

Réussir à

A mon humble avis

OBLIGATION ⇔ a

Structure

Obligation **d'**une personne **par** une autre **à** faire quelque chose / d'un objet.

[p]	of	personne obligée
[p]	by	personne obligeant
[a]	todo	action sur laquelle porte l'obligation
[o]	object	objet obligé

Commentaires

-

Exemples

Falloir

Devoir

Etre forcé/obligé/indispensable/obligatoire/interdit de/défendu de/se voir contraindre (à/de)

Défendre de

Obliger à

Interdire de

Forcer à

WILL

Structure

Volonté **de** quelqu'un **qu'**une action soit réalisée **par** une personne / d'un **objet**.

[p]	of	personne demandeuse
[p]	by	personne devant réaliser l'action
[a]	todo	action à réaliser
[o]	object	objet demandé
[ti]	instant	instant où l'on désire que l'action soit effectuée

Commentaires

-

Exemples

Ça te dirait de
Tu accepterais de
J'aimerais
Si seulement...
Souhaiter
Désirer
Être souhaitable/désirable/mieux
Être désireux de
Vouloir
Refuser / Rejeter
Avoir besoin de
Attendre de + acteur + que
ne pas demander mieux que
être reconnaissant si
suggérer
se décider pour
c'est avec grand plaisir que
Laisse moi + action
J'aimerais bien
Prendre la décision de

PROBABILITY

Structure

Probabilité **d'**une action / **d'**un objet à un certain **degré**.

[a]	of	action modalisée
[o]	of	objet modalisée
[[0,4]]	degree	degré de probabilité de l'action (0=nul, 1=peu probable, 2=probable , 3=très probable, 4=certain)

Commentaires

Il existe une forte ambiguïté difficile à lever sur "pouvoir", qui peut avoir valeur de possibilité (épistémique) ou de probabilité (déontique).

Exemples

Se pouvoir que
Il n'y a aucune chance que
Il n'y a pas l'air de
Il est fort probable que
Il n'est guère probable que
Certain que
On m'a dit que
Ne faire aucun doute que
Avoir dû + action

EXISTENCE ⇔ ○

Structure

Existence **de** quelque chose **dans** un endroit.

[o]	of	objet dont on considère l'existence Recouvre également les modalités de type assimilable à object, comme WAY, DIFFERENCE, LIMIT, VALUE...
[t]	time	temps dans lequel on considère l'existence
[l]	in	lieu où l'on recherche l'existence

Commentaires

-

Exemples

Il (n')y a (ou pas) / (n')y a-t-il

Est-ce que / si + lieu/contexte + avoir/posséder + object

Il existe / Existe-t-il

Est-ce que + DIFFERENCE

Manquer

OTHER ⇔ a / o

Structure

Un objet / Une action autre **que** celui / celle considéré(e).

[o]	than	objet considéré
[a]	<i>than</i>	action considérée

Commentaires

OTHER est un schéma transparent, qui prend la nature (action ou objet) de l'entité du champ than.

Exemples

Autre chose que + objet / action

OBJECT ⇔ o

Structure

Un **objet** donné ayant un **type**, des **propriétés** et situé en un **lieu** donné. Il peut servir à **effectuer** une action ou **être objet** de celle-ci.

[o]	object	nom de l'objet, notamment utile si l'on ne sait pas extraire les informations ci-dessous
[t]	type	type de l'objet
[P]	property	propriété vérifiée par l'objet
[a]	doing	action effectuée par l'objet
[a]	subject-of	action dont l'objet est objet
[l]	place	espace auquel appartient l'objet
[t]	time	instant où une action liée à l'objet est effectuée
bool	multiple	l'objet recherché est unique (0) ou non (1)

Commentaires

C'est une entité un peu passe-partout, dont le cadre d'utilisation n'est pas facilement définissable.

Il peut être ardu de différencier une action faite par l'objet (doing) d'une action dont l'objet est objet (subject-of). De plus, il n'est pas forcément facile de mettre un nom sur l'action en question à partir d'une description ou d'un terme approximant.

Exemples

DIFFERENCE ⇔ ○

Structure

Différences **entre** une entité **et** une autre **selon** un critère.

[o]	<i>between</i>	Premier objet considéré
[c]	<i>between</i>	Premier concept considéré
[P]	<i>between</i>	Première propriété considérée
[a]	<i>between</i>	Première action considérée
[o]	<i>and</i>	Second objet considéré
[c]	<i>and</i>	Second concept considéré
[P]	<i>and</i>	Seconde propriété considérée
[a]	<i>and</i>	Seconde action considérée
	<i>according</i>	Critère de différenciation considéré
	<i>is</i>	Différence constatée

Commentaires

Difficulté de typer "according" et "is"...

Il faudrait éventuellement considérer d'utiliser d'autres modalités selon ce champ, mais elles n'existent pas toutes.

Par exemple :

```
CHECK(NEG(DIFFERENCE(between="le bouton 'fermer'", and="le bouton 'quitter'",  
is="fonctionnement"))))
```

Pourrait s'écrire :

```
CHECK(NEG(DIFFERENCE(between=FUNCTIONING(of="le bouton 'fermer'"),  
and=FUNCTIONING(of="le bouton 'quitter'"))))
```

Mais ça ne fonctionne pas pour :

```
ASK(NEG(DIFFERENCE(between="touche entrée du clavier",  
according="fonctionnalités"))))
```

Exemples

Objet1 + avoir le même + critère + que + objet2

Objet1 + et + objet2 + avoir le même + critère

Objet1 + et + objet2 + sont différents/liés/identiques

La différence entre + objet1 + et + objet2 + c'est + différence

Ce qui distingue + objet1 + et + objet2 + c'est + différence

Objet1 + et + objet2 + marcher en sens inverse

Propriété1 + alors que + Propriété2

Revenir au même + action1 + que + action2

Une différence/ressemblance entre + objet1 + objet2

ORDER ⇔ a

Structure

Réalisation d'un ensemble d'actions **étape** par **étape**, la dernière d'entre elles dans l'ordre d'énumération étant considérée comme le but de la procédure.

[a]	step1	1 ^{ère} action à réaliser
[a]	step2	2 nd e action à réaliser
[[1,2]]	goal	numéro de l'étape correspondant à l'élément principal de la relation d'ordre s'ils sont liés

Commentaires

Selon les cas, le focus n'est pas sur le même élément : on peut s'intéresser à la première action, à la dernière, ou à la procédure prise dans son ensemble ! Le problème est que ce but ne peut pas être entièrement déterminé a priori : on peut savoir si le focus au sein de la relation d'ordre porte sur le premier ou le dernier élément, mais pour savoir s'il constitue réellement un but, il faut considérer le cadre entourant le schéma ORDER().

1. est-ce qu'on est obligé de stopper avant de redémarrer ?

Step1 = Stopper()

Step2 = Redémarrer()

But = Redémarrer()

ASK(OBLIGATION(ORDER(1,2,goal=2))) → Utilité de 1 ?

1b. est-ce que c'est vraiment indispensable d'afficher une fenêtre popup pour modifier une simple ligne ?

Step1 = Afficher("pop-up")

Step2 = Modifier("ligne")

But = Modifier("ligne")

ASK(OBLIGATION(ORDER(1,2,goal=2))) → Utilité de 1 ?

2. Est-ce qu'on doit cliquer sur OK après (avoir redémarré / redémarrer) ?

Step1 = Redémarrer()

Step2 = Cliquer(OK)

But = Redémarrer()

ASK(OBLIGATION(ORDER(1,2,goal=1))) → Utilité de 2 ?

3. où ça mène de stopper puis de redémarrer le compteur?

Step1 = Stopper()

Step2 = Redémarrer()

But = N/A, à cause du type de la question qui s'intéresse à la procédure dans son ensemble.

ASK(EFFECT(ORDER(1,2))) → Effet de la combinaison (1,2)

4. quand dois-je appuyer sur "stop" pour arrêter exactement à 100

Step1 = Cliquer("stop")

Step2 = Arrêter("à 100")

But = Arrêter("à 100")

ASK(MOMENT(ORDER(1,2,goal=2))) → Moment où faire 1 ?

5. Doit-on cliquer sur le bouton reset avant le bouton démarrer pour lancer le compteur ?

Step1 = Cliquer("reset")

Step2 = Cliquer("démarrer")

Step3 = Démarrer("compteur")

But = Démarrer("compteur")

ASK(OBLIGATION(ORDER(1,2,3,goal=3))) → Utilité de 1 pour 2 ? Utilité de 2 pour 3 ?

On peut alors préférer se limiter à 2 étapes, quitte à imbriquer, ce qui donnerait ici :

Step1 = Cliquer("reset")

Step2 = Cliquer("démarrer")

But = Cliquer("démarrer")

Step1 = Cliquer("démarrer")

Step2 = Démarrer("compteur")

But = Démarrer("compteur")

ASK(OBLIGATION(ORDER(ORDER(1,2,goal=2),3,goal=3)))

Mais la syntaxe n'est pas extrêmement claire ainsi.

Néanmoins, on notera que le cas où la relation d'ordre met en jeu plus de 2 éléments se produit rarement (au point qu'elle n'apparaît même pas dans le corpus, la phrase d'exemple ci-dessus étant construite).

En conclusion, on retiendra de manière générale que l'élément but de la relation d'ordre est celui qui se situe après les marqueurs d'ordre (avant, après, pour...), en dernier dans l'énumération, et l'on s'intéresse alors à la prémisse (exemples 1, 1b et 2). Dans le cas de "puis" en revanche, il n'y a pas de notion de but et l'enchaînement des actions est à prendre tel quel.

Exemples

De + action1 + avant/puis de/pour + action2

Action1 + d'abord avant de + action2

WAY ⇔ o

Structure

Moyen d'atteindre un **but**.

[a]	goal	But à atteindre
{}	qual	Qualificatif concernant le moyen recherché {"none", "best", "another"}

Commentaires

-

Exemples

Comment / Comment faire (pour)

Que (falloir) faire (pour) / Quoi faire (pour) / De quelle façon + action

Un moyen de / la façon de / un tuyau pour / la marche à suivre pour + action

La meilleure solution pour + action

Les conditions pour + action

EFFECT ⇔ ○

Structure

L'effet **d'**une action / d'un objet **est** une autre action.

[a]	of	action dont on considère l'effet (cause)
[c]	<i>of</i>	concept dont on considère l'effet
[o]	<i>of</i>	objet dont on considère l'effet
[a]	is	action correspondant à l'effet de l'action 'of'

Commentaires

-

Exemples

Ce qui arrive/se passer si/quand + action-of

Effet/impact (que) (ça fait/a de) + action-of

Sur quoi débouche + action-of

action-is + si/quand + action-of

action-of + pour + action-of

Où ça mène de + action-of

action-of + provoquer + action-is

REASON ⇔ o

Structure

La raison/cause **de** quelque chose.

[a]	<i>of</i>	action dont on considère la raison
[c]	<i>of</i>	concept dont on considère la cause
[o]	<i>of</i>	objet dont on considère la cause

Commentaires

-

Exemples

le fait que + ...

pourquoi + ...

pour quelle raison + ...

qu'est-ce qui fait (que) + ...

parce que + ...

car + ...

PLACE ⇔ o

Structure

Emplacement **de** quelque chose / **où** faire quelque chose.

[o]	of	objet dont on considère l'emplacement
[p]	of	personne dont on considère l'emplacement
[a]	where	action dont on cherche à savoir où elle peut être réalisée

Commentaires

Schéma parfois implicite et difficile à détecter en l'absence de "où".

Exemple : "je ne trouve pas X" → NEG(KNOWLEDGE(of=u, about=PLACE(of="X")))

Exemples

Où pouvoir être / trouver
(ne pas trouver) + objet

MOMENT ⇔ o / t

Structure

Instant **où** une action se fait / **où** un événement a lieu.

[a]	when	action dont on considère l'instant de déroulement
[o]	when	un événement dont on considère l'instant de déroulement (EXISTENCE, VALUE, ...)
[t]	time	instant passé / présent / futur

Commentaires

Schéma parfois implicite et difficile à détecter en l'absence de "quand".

Exemple : "je veux que le compteur stoppe à 1000"

→ WILL(of=u, todo=Arrêter(o="compteur"), time=MOMENT(when=VALUE(o="compteur", is="1000")))

Exemples

Quand

Au moment où

A l'instant où

Lorsque

Que quelque chose stoppe (quand)...

NUMBER

Structure

Nombre **d'**éléments **dans** un espace donné.

[o]	of	objets à dénombrer
[l]	in	espace où l'on dénombre

Commentaires

-

Exemples

Combien de + object

LIKE

Structure

Un **agent** aime un **objet aimé** (quelqu'un ou quelque chose).

[p]	agent	personne qui aime
[p]	liked	personne aimée
[o]	<i>liked</i>	objet aimé
[c]	<i>liked</i>	concept aimé

Commentaires

-

Exemples

aimer + objet

préférer + concept

emballer

objet + être (un peu/très) ridicule/moche

FEAR

Structure

Un **agent** craint une **crainte**.

[p]	agent	personne qui craint
[o]	fear	objet craint
[a]	<i>fear</i>	action crainte

Commentaires

-

Exemples

avoir [bien] peur [de/que] + crainte

BOTHER

Structure

Le **gêné** est incommodé par une **gêne**.

[p]	embarrassed	personne gênée
[o]	embarrassment	objet gênant
[a]	<i>embarrassment</i>	action gênante

Commentaires

-

Exemples

ennuyer [de/que] + gêne

déranger [de/que]+ gêne

DOUBT

Structure

Un doute **d'**une personne **au sujet de** quelque chose.

[p]	agent	personne doutant
[o]	about	objet sur lequel porte le doute

Commentaires

Possible chevauchement entre DOUBT() et PROBABILITY() : douter de quelque chose c'est considérer qu'une chose est improbable...

Exemples

Avoir un/des doute(s) au sujet de + object

Object + étonner + person

Etre sûr de + object

Ah bon ?

Vraiment ?

SURPRISE

Structure

Un **agent** est surpris **par** quelque chose.

[p]	agent	personne surprise
[o]	about	objet surprenant

Commentaires

-

Exemples

Surprendre / étonner

Etre bizarre / surprenant / étonnant / étrange + objet

Etre (vraiment) surpris / étonné que + objet

REGRET

Structure

Un **agent** a des regrets **au sujet de** quelque chose.

[p]	agent	personne regrettant
[o]	about	chose regrettée

Commentaires

-

Exemples

Regretter de + objet

Etre dommage/embêtant que + objet

HAPPY

Structure

Un **agent** est heureux **au sujet de** quelque chose.

[p] agent
[o] about
[a] *about*

Commentaires

-

Exemples

Se désoler que + objet
être déçu que + objet

PROBLEM ⇔ o

Structure

Situation non satisfaisante, quelque chose ne se passe pas comme attendu.
Problème **d'**une personne **avec** un objet ou **pour** faire une action.

[p]	of	personne ayant le problème
[o]	with	objet problématique
[a]	for	action problématique
[t]	time	moment où l'action a posé / pose problème

Commentaires

-

Exemples

avoir du mal à + action
problème pour + action
être difficile de + action
problème de + objet
être buggé / être mal programmé + objet
objet + [ne pas fonctionner/marcher] [comme il faut/normalement]
objet + planter
objet + être incorrect
[quelque chose/ça] ne va pas avec + objet
objet + en panne / fou
un bug avec + objet

MISTAKE ⇔ o

Structure

Acte involontaire/inadapté, généralement au passé (mais ça peut être une crainte d'une erreur à venir).

Un **responsable** commet une erreur **sur** une **action** ou un **objet**.

[p]	responsable	personne responsable de l'erreur
[o]	about	objet erroné
[a]	<i>about</i>	action erronée
[t]	time	temps de l'erreur (passé = déjà faite, présent = ...)

Commentaires

« Il y a une erreur » (résultat d'une action passée)

« Je crains de faire une erreur » (résultat d'une action potentielle à venir)

Exemples

action + "sans le faire exprès"

erreur action

"se tromper de" + objet

erreur objet

"se tromper"

erreur

"avoir fait une bêtise"

erreur

"mal faire"

erreur

HELP ⇔ o / a

Structure

Un **demandeur** réclame de l'aide à un **assistant**, (**pour** effectuer une action / **au sujet de** quelque chose), **en** faisant quelque chose (éventuellement).

[p]	seeker	personne sollicitant l'aide
[p]	helper	personne dont l'aide est requise
[a]	for	action requérant de l'aide, but à atteindre
[a]	by	action à faire pour aider
[o]	about	chose pour laquelle il faut de l'aide

Commentaires

La nuance entre "pour" et "en" n'est pas toujours évidente.

Ex : "m'aider pour modifier la 2^e ligne" → objectif : modifier la ligne, moyen : la modifier toi-même ou m'expliquer comment faire ?

On peut éventuellement considérer qu'en l'absence de 2 champs clairement distincts, ils sont égaux.

Exemples

Aide moi à

M'aider [|à|pour]

Ne me laisse pas tomber

M'apporter ton soutien pour

Je t'aide

M'aiderait que

Te demander de l'aide

Compter sur ton aide

Faire le prof

Me conseiller

Résoudre

TELL ⇔ a

Structure

Un **locuteur** s'adresse à un **interlocuteur** pour lui parler d'un **sujet** ou lui exposer une **action** à réaliser.

[p]	speaker	personne disant
[p]	interlocutor	personne à qui il est dit
[o]	<i>interlocutor</i>	objet à qui il est dit
[o]	about	objet dont il est objet
[a]	<i>about</i>	action dont il est objet
[a]	todo	action qui doit être réalisée par l'interlocuteur
[t]	time	temps de narration

Commentaires

-

Exemples

Explique moi

Dire à Coco d'arrêter

Dire ça

Tu as dit

[Que/si] tu me [dises/disais] [si/comment/quoi]

Décrire

Prévenir

Rappeler

Annoncer

CHECK ⇔ a

Marqueur de demande de vérification d'une information : il peut s'agir de vérifier qu'une action a bien été réalisée, que l'on parle bien d'une même entité, ou qu'un élément est bien comme on l'imagine (il est placé complètement à l'extérieur de la requête et ne peut être imbriqué).

Ça... c'est ça ?	CHECK(action)
Est-on bien sûr que ...	CHECK(action)
Objet être (bien) de type X	CHECK(TYPE(objet="...", type=X))
Objet avoir valeur X	CHECK(VALUE(objet="...", value=X))
X ? (ex : "les boutons du milieu ?")	CHECK(objet="X")
Etre assuré que ...	CHECK(action)

ASK

Marqueur d'interrogation, pour toute interrogation qui n'est pas un CHECK, c'est-à-dire dans laquelle on ne présuppose pas du résultat de la question (il est placé complètement à l'extérieur de la requête et ne peut être imbriqué). Par exemple :

Objet être (bien) de type X	CHECK(TYPE(objet="...", type=X))
Quel est le type de objet	ASK(TYPE(objet="..."))

NEG

Marqueur de négation d'une action ou d'une modalité : il peut potentiellement précéder n'importe quel schéma et peut donc se trouver à n'importe quelle couche dans l'imbrication. Sa position dans la requête formelle est à prendre en compte pour en retrouver le sens. Ainsi :

Tu dois modifier Coco	OBLIGATION(Modifier(o= « Coco »))
Je t'interdis de modifier Coco	OBLIGATION(NEG(Modifier(o= « Coco »)))
Je ne t'interdis pas de modifier Coco	NEG(OBLIGATION(Modifier(o= « Coco »)))

OBLIGATION(NEG(A())) = interdiction

NEG(OBLIGATION(A())) = absence d'obligation

De la propagation des marqueurs

Lorsqu'un schéma externe contient un champ time renseigné, il se propage implicitement aux autres champs time des schémas qu'il contient.

Par exemple : "Sais-tu quand il y aura des pages en anglais ?"

```
ASK(KNOWLEDGE(of=a, about=MOMENT(time=4,  
EXISTENCE(of=OBJECT(type="pages", P="en anglais")))))
```

→

```
ASK(KNOWLEDGE(of=a, about=MOMENT(time=4, EXISTENCE(time=4,  
of=OBJECT(type="pages", P="en anglais")))))
```