III.4. Explication et critique argumentatives dans les systèmes à base de connaissances

1. Introduction

L'utilisation des systèmes à base de connaissances (SBC)[(1)] rencontre des problèmes qui semblent liés à des limitations non des systèmes eux-mêmes, mais de leurs utilisateurs ! En effet, les SBC ont souvent la capacité théorique de produire des raisonnements qui pourraient être pertinents pour leur utilisateur à un moment donné. Malheureusement, soit le raisonnement pertinent n'est pas produit, parce que l'utilisateur ne sait pas exprimer son besoin dans des termes assimilables par le système, soit le raisonnement pertinent est effectivement produit, mais l'utilisateur est incapable de le comprendre à travers le résultat que lui fournit le système.

Il s'agit là d'un problème d'*explication*. Lorsque les utilisateurs font appel au SBC, ils expliquent (= exposent) leur problème d'une façon qui n'est pas nécessairement adaptée à ce que le système est capable de reconnaître. De même, lorsque le SBC leur fournit la trace d'un raisonnement (enchaînement de règles), ils ne sont pas en mesure de reconnaître l'explication (= solution) de leur problème.

Il est à peu près clair maintenant que la meilleure acceptabilité des SBC, qui est fortement attendue, passera par leur adaptation à l'utilisateur humain, puisque l'adaptation inverse semble impossible. Mais, comme nous le verrons, les solutions proposées divergent quelque peu.

Puisqu'il s'agit d'amener les SBC à communiquer d'une manière humainement acceptable, il est légitime de penser que notre étude de certaines des contraintes cognitives qui régissent les interactions spontanées entre humains doit pouvoir apporter un nouvel éclairage sur ce problème complexe d'interaction humain / machine.

Nous suggérons ainsi, comme d'autres, le fait que cette interaction doit prendre la forme d'un dialogue. Notre originalité consiste à dire que ce dialogue doit ressembler à une *conversation*, au sens restreint de ce terme, c.à.d. qu'elle doit reposer sur une *argumentation critique*.

A cet égard, nous nous rapprochons de certains travaux qui visent à concevoir des "Systèmes Experts Critiques". De tels systèmes ont pour fonction de critiquer des éléments de solution fournis par l'utilisateur, plutôt que de proposer des solutions globales peu assimilables. Par rapport à ces recherches, notre originalité consiste à proposer les critiques sous forme d'une argumentation.

Nous nous situons donc à l'intersection de deux courants : celui qui veut amener les SBC à gérer l'*explication* sous la forme d'un dialogue, et celui qui conçoit le rôle du SBC comme celui d'un *critique*. Notre contribution tire son originalité de la compréhension des phénomènes d'explication dans les relations conversationnelles.

---

[1] Cette dénomination tend à remplacer la dénomination, plus restrictive et souvent mal perçue, de *système expert*.

2. La problématique de l'explication dans les SBC

2.1. Ajouter des connaissances spécifiques pour la génération d'explications

Face à l'incapacité des systèmes à base de connaissances à *expliquer* leurs raisonnements de manière à les faire accepter, de nombreux auteurs ont incriminé l'absence dans le système de certaines connaissances nécessaires à la production d'explications. Les SBC disposent bien entendu de connaissances qui leur permettent de mener à bien leurs raisonnements, mais on s'est vite aperçu que la simple trace du raisonnement ne constituait pas une explication acceptable. Des auteurs comme W.J. Clancey [1981] et W.R. Swartout [1983] ont insisté sur la nécessité de doter le système de moyens pour analyser son propre raisonnement.

Clancey a doté son système NEOMYCIN de divers types de connaissances qui sont venues s'ajouter aux règles d'inférence du système expert MYCIN, notamment des règles stratégiques capables de justifier le déclenchement préférentiel de certaines règles d'inférence dans certaines situations. Ces règles stratégiques étaient inutiles lors de la production du raisonnement (dans le cas de MYCIN, il s'agissait de déterminer un agent infectieux responsable de maladies comme la méningite), mais elles servaient à expliquer le raisonnement (dans le cas de NEOMYCIN, il s'agissait d'enseigner, cf. [Clancey 1987]).

Alors que Clancey a dû adjoindre *a posteriori* les connaissances nécessaires à la génération d'explications, Swartout juge quant à lui plus "économique" de ne pas *perdre* ces informations. En effet, les connaissances qui permettront d'expliquer sont là au départ : ce sont les connaissances du domaine, et elles sont explicites lors de la conception du système. Malheureusement, elles sont généralement perdues lors de la "compilation" (pour reprendre un terme de Clancey) de la connaissance sous forme de règles d'inférence. Swartout prend l'exemple de la règle d'inférence suivante :

> "Si on observe une baisse de potassium dans le sérum alors que de la digitaline a été administrée, alors baisser la dose de digitaline."

En fait, ce type de règle dérive d'une connaissance plus profonde, qui précise que la baisse du taux de potassium, associée à une dose trop élevée de digitaline, peut entraîner une fibrillation ventriculaire. Cette connaissance dérive elle-même d'un schéma plus abstrait, qui peut s'exprimer ainsi :

> modèle : "toxicité d'une drogue" - Si une anomalie (*e.g.* anomalie de dosage) est associée à une augmentation de la dose d'une drogue, cela peut entraîner un dérèglement dangereux.

En décrivant son système XPLAIN, Swartout insiste sur l'intérêt de conserver toutes ces informations lors de la création du programme expert. Cela permet au système de fournir des explications comme celle-ci :

> Please enter the value of serum potassium : **`why ?`**
>
> The system is anticipating digitalis toxicity. Decreased serum potassium causes increased automaticity, which may cause a change to ventricular fibrilation. Increased digitalis also causes increased automaticity. Thus, if the system observes decreased serum potassium, it reduces the dose of digitalis due to decreased serum potassium.

L'adjonction au SBC de connaissances spécifiques aux besoins de l'explication constitue un progrès significatif. Grâce elles, des systèmes comme NEOMYCIN ou XPLAIN sont capables d'*expliquer* leur comportement en termes de buts et de démarche. Pourtant, quoique nécessaire, cette solution n'est peut-être pas suffisante.

2.2. L'explication émergeant d'un dialogue

B. Safar [Safar et al. 1992] souligne le fait que dans de nombreuses applications, les représentations internes du système et les détails de sa démarche n'intéressent pas l'usager, et ne constituent donc pas une explication. Nous tournons là le dos à "l'explication-trace" (*"translate-the-code approach"*). De même, C. Paris [1992] constate que les modules d'explication sont généralement ajoutés après coup aux SBC, et fournissent des explications dérivées de motifs *ad hoc*, alors que l'on peut concevoir que la résolution de problème soit indissociable du processus explicatif lui-même : le SBC est alors perçu, non plus comme un

système chargé de résoudre un problème, mais comme un système chargé d'*expliquer* (un problème, une solution, un phénomène). On évolue ainsi inévitablement vers une conception *dialogique* de l'explication.

En effet, l'idée d'une explication dynamique constituant la totalité de l'interaction, par opposition à une phase explicative qui viendrait seulement après une phase de résolution de problème, commence à s'imposer pour plusieurs raisons :

- l'usager est rarement capable d'exprimer ses buts d'une manière précise et complète
- le système ne peut pas satisfaire tous les buts de l'usager
- la satisfaction complète ou incomplète de certains buts peut engendrer de nouveaux buts de la part de l'usager

Ces raisons amènent à concevoir l'explication comme une *négociation*, caractéristique de certains types d'interactions dialogiques [Baker 1992]. Les phases de raisonnements, liées à la satisfaction de buts partiels, sont ainsi mêlées aux phases d'analyse de la requête et de propositions de compromis.

La problématique de l'explication a ainsi complètement bouleversé la conception du rôle du SBC. Perçu au départ comme un résolveur de problème, le système à base de connaissances devient avant tout un système capable de fournir des explications adaptées au besoin de l'utilisateur. Les capacités de raisonnement du SBC ne sont plus mises en avant, elles sont mises au service de l'explication [GENE 1993].

Dans cette conception, c'est tout de même le système qui résout le problème de l'usager, ou un problème approchant, de manière à satisfaire l'usager. Dans l'approche des systèmes critiques, à l'inverse, c'est l'usager qui résout lui-même son problème : le système va simplement l'aider dans sa tâche en émettant des critiques appropriées.

2.3. L'explication vue comme une critique des actions de l'utilisateur

La motivation qui a présidé à la création des systèmes critiques trouve aussi son origine dans le souci de changer le rôle du SBC pour qu'il soit mieux accepté. Le système à base de connaissances, utilisé comme un critique, n'est plus chargé de résoudre un problème. Il est engagé dans une relation de *coopération* avec l'usager :

> "Cooperative problem-solving systems are examples of human-computer cognitive systems. They serve as cognitive amplifiers of the human. [...] The major difference between classical expert systems, such as MYCIN and R1, and cooperative problem-solving systems involves the roles of the user and computer. Most expert systems ask the user for input, make all decisions and then return an answer. In a cooperative problem-solving system, the user is an active agent empowered by the system's knowledge." [Fischer et al. 1991]

L'usager du système est engagé dans une tâche, et entreprend des actions pour la résoudre. Le système critique observe ces tentatives, et intervient lorsqu'il est en mesure d'émettre une critique utile. Pour cela il évalue les actions de l'usager en fonction des buts poursuivis (but de la tâche ou objectifs analysés comme des sous-buts de l'usager), il envisage des solutions alternatives moins coûteuses, il teste l'éventuelle violation de contraintes par l'action entreprise par l'usager (contraintes de validité et contraintes de préférence). Cette démarche est en accord avec ce qu'indique l'observation des concepteurs (*e.g.* ingénieurs) en situation de résolution de problème :

> "L'élaboration de la solution s'effectue en interaction continue avec l'évaluation de la solution. La solution finale est élaborée progressivement, par adjonction d'éléments de solution successifs. Avant d'adopter un élément de solution, le concepteur l'évalue. Cette évaluation aboutit à un constat de respect - ou non - des contraintes fixées. Il en résulte un jugement d'acceptabilité de l'élément de solution et une orientation de l'attention du concepteur vers des éléments de solution plus spécifiques." [Bonnardel 1989].

Cette approche par la critique est donc conçue pour apporter une aide pertinente à l'usager, comme dans le cas des SBC conçus pour l'explication. L'ambition est toutefois en apparence plus limitée. Il ne s'agit pas de

fournir une solution complète à l'usager, à la suite d'une éventuelle négociation du problème. Il s'agit simplement de donner ici et là des critiques pertinentes.

Pourtant, on peut voir là l'un des points forts des systèmes critiques. Ils n'ont pas besoin de posséder une expertise complète du domaine. Ils peuvent fonctionner utilement quel que soit leur niveau de compétence, en n'émettant que les critiques qu'ils ont réussi à calculer. Les systèmes critiques peuvent ainsi se contenter d'une expertise partielle, alors que les systèmes "experts" ne fonctionnent correctement que dans des situations où ils sont parfaitement compétents.

Notre observation des conversations et du contenu des répliques émises par les interlocuteurs nous amène à justifier une approche qui combine les avantages des deux courants que nous avons évoqués. En effet, les explications humaines d'une part sont dialogiques et nécessitent parfois une conversation entière, et d'autre part elles prennent la forme d'une critique entre des acteurs ayant un niveau d'expertise comparable sur le sujet de la conversation.

Nous proposons de caractériser l'explication telle quelle apparaît dans les conversations, puis de voir comment elle peut être transposée dans des situations de résolution de problème entre un humain et un système à base de connaissances.

3. Anatomie de l'explication conversationnelle

3.1. Définir l'explication

L'explication dans les SBC est un concept qui, à notre avis, regroupe des problématiques différentes. Le mot *explication* a trois sens principaux :

- on peut expliquer un terme obscur (par ex. le mot *rhombe*)
- on peut expliquer par un long développement (par ex. la théorie de la relativité restreinte)
- on peut expliquer un phénomène étonnant (par ex. "*de quoi est-il mort si jeune ?*")

Ces acceptions correspondent toutes trois à des besoins réels des SBC. Toutefois, nous défendons l'idée que ces besoins ne seront pas satisfaits par les mêmes moyens. Nous choisissons de nous intéresser ici exclusivement au troisième sens de l'explication, celui qui est relatif à l'étonnement, et ceci pour trois raisons.

Tout d'abord, c'est celui pour lequel l'étude des conversations nous donne le meilleur éclairage. De plus, il correspond, dans le contexte des SBC, au besoin le plus difficile à satisfaire, alors que des moyens classiques (e.g. hypertextes) semblent pouvoir apporter une aide significative pour l'explication de termes obscurs ou de chapitres entiers de la connaissance (cf. notre expérience avec SAVANT1 [Dessalles 1984]). Enfin, nous pensons qu'il est possible d'utiliser les principes de l'explication basée sur la surprise non seulement pour satisfaire les requêtes ponctuelles des usagers ou pour les aider dans une résolution de tâche, mais aussi pour engendrer des développements plus longs destinés à leur *expliquer* (dans les deux premiers sens du mot) des termes ou des phénomènes (cf. chap. IV.1).

3.2. L'explication après surprise

Le phénomène de l'explication est très fréquent dans les conversations et dans la vie courante. Prenons l'extrait suivant, dans lequel le mot *explication* est explicitement mentionné.

[télévision 21-02-92]

commentateur A- Formidable Surya Bonaly, qui vient de réaliser, pour la première fois dans l'histoire du patinage, une qua-dru-ple boucle et qui ne s'est pas contentée de cela. Elle a passé [... quatre triples sauts]

[...] [les notes techniques s'affichent]

commentateur B- Quelle déception. Quelle déception.

commentateur A- *Oh la la la la la la la la la la la!*
commentateur B- *Quelle déception!*
commentateur A- *5.1 de la part de la juge anglaise! Alors là, il y a une chose que l'on a du mal à comprendre.*
[silence 4 secondes]
commentateur B- *L'explication vient peut-être de la réception des sauts.*
[silence 6 secondes]
commentatrice - *de la réception et aussi de la préparation. C'est-à-dire qu'il y a toujours un petit blocage dans sa préparation.*

On perçoit clairement ici comment l'explication vient résoudre un étonnement ("*il y a une chose que l'on a du mal à comprendre...*"). Cet étonnement vient de la contradiction entre les prouesses techniques réalisées par S. Bonaly et la faiblesse de la note technique octroyée:

[prouesses_techniques **&** note_technique_faible] $\Longrightarrow$ **F**

Mais l'explication (noter le mot "explication" spontanément employé par le deuxième commentateur) est produite ici par une **invalidation indirecte** (cf. chap. I.1) qui infirme une prémisse absente au départ. En effet, dans notre représentation, la formule précédente doit être remplacée après les deux dernières répliques par quelque chose comme:

[prouesses_techniques **& non** défaut_majeur **&** note_technique_faible] $\Longrightarrow$ **F**

L'incompatibilité de départ n'a donc plus cours, car une prémisse insatisfaite avait été omise. Mentionner *a posteriori* quelque chose qui apparaît comme un défaut majeur, apparaît ainsi pour les commentateurs comme un moyen de résoudre la contradiction.

Le fait que le type d'explication auquel nous nous intéressons soit défini par rapport à une incohérence, source de l'étonnement, fait de ce genre d'explication un phénomène fortement contraint. Si on exprime la contradiction de référence de la manière suivante:

[p_1 **&** p_2 **& . . . &** p_n] $\Longrightarrow$ **F**

alors une explication *q* est admissible si et seulement si:

- soit q = ***non*** p_i, ou bien q permet de démontrer ***non*** p_i (invalidation directe)
- soit $[\ p_1$ **&** p_2 **& . . . &** p_n **&** ***non*** $q\ ]\ \Rightarrow$ ***F*** peut être acceptée par l'interlocuteur comme étant la règle correcte (invalidation indirecte)

Nous avons vu des exemples d'invalidation directe dans le chapitre I.1. Nous avons vu aussi que la distinction entre invalidations directe et indirecte, dans certains cas limites, pouvait dépendre de la représentation logique. Le point fondamental à retenir est que l'explication conversationnelle se traduit par la *suppression d'un étonnement* : soit par la réfutation de l'un des termes explicites de l'étonnement, ou tout au moins sa mise en doute (cf. le prochain extrait), soit par la réfutation d'un terme "oublié" (*i.e.* une hypothèse implicite).

L'invalidation indirecte peut être considérée à juste titre comme "plus riche", dans la mesure où elle amène des éléments nouveaux. Il est important de noter que ce qui la caractérise, c'est qu'elle mentionne un fait non seulement absent de la contradiction initiale, mais surtout qui puisse être **reconnu** et accepté par le premier locuteur comme une prémisse oubliée. C'est ce qui en fait une opération fortement contrainte, au point qu'il est envisageable qu'un système artificiel puisse reconnaître et émettre des invalidations même indirectes : soit une prémisse est volontairement omise par le système dans un premier temps (comme dans le cas de SAVANT3, cf. chap. III.2), l'explication étant ensuite reconnue comme une invalidation de cette premisse ; soit le système dispose de deux niveaux de connaissance d'accessibilité différente (cas de l'EBL), et une contradiction au niveau superficiel peut disparaître ensuite lors de l'accès au niveau profond, grâce à la présence de prémisses non satisfaites (cf. discussion dans le chap. I.3).

3.3. L'explication par explicitation

Les arguments précédents laissent supposer que toute explication (au sens où nous l'entendons, *i.e.* excluant les définitions de termes et les longs développements) pouvait être traduite par l'invalidation d'un étonnement. Ce serait oublier le cas de l'*explicitation*, qui consiste au contraire à expliciter l'étonnement lui-même (ou plus précisément la clause de référence, cf. chap. I.1 et I.3) :

A1: Tu es bien du Sud-ouest? Est-ce que la Dordogne fait partie de la Gascogne?
B1: Non. Ca fait partie de l'Aquitaine, mais la Gascogne, c'est plus au sud.
A2: Parce que dans [le film] *"Cyrano de Bergerac", ils présentent Cyrano comme un Gascon.*
B2: Ah, mais Bergerac, c'est différent. Ca ne fait pas partie du Périgord. Le Périgord n'est pas en Gascogne. Mais le Bergeracois, peut-être.

La réplique A2 est indéniablement une explication (noter le *parce que* qui l'introduit). Ce n'est pourtant pas l'invalidation, mais l'explicitation d'un contexte incohérent. Bergerac étant en Dordogne, A s'étonne de ce que Cyrano soit qualifié de gascon. Il y a contradiction logique entre le fait d'appartenir à la Dordogne et le fait d'être qualifié de gascon, sachant que pour A, la Dordogne est hors de la Gascogne.

En revanche, B2 est bien une explication (ou une tentative d'explication) qui agit comme une invalidation : B2 a pour effet de supprimer, d'invalider le caractère incohérent du contexte :

notations	*contexte*
cb: Cyrano est de Bergerac	[bd **&** ***non*** dg **&** bg] ⇒ ***F***
dg: la Dordogne est en Gascogne	[cg **&** cb] ⇒ bg
bd: Bergerac est en Dordogne	cb
cg: Cyrano est gascon	***non*** dg
bg: Bergerac est en Gascogne	bd
	cg

Les propositions *bd*, *bg* et [***non*** *dg*] sont incompatibles, or elles se retrouvent toutes vraies. La première clause du contexte est alors saturée (*i.e.* tous ses termes sont vrais) et il y a donc bien une incohérence, qui pousse A à poser la question A1. L'explication A2 est une *explicitation* du contexte problématique via la deuxième clause.

B1 vient confirmer [***non*** dg]. En revanche B2 est bien une explication invalidante. Nous pouvons modéliser le raisonnement de B en notant qu'il substitue le Périgord à la Dordogne dans le raisonnement de A. Il refuse donc la première règle(2), et il la remplace par (en notant *bp* : Bergerac est en Périgord, et *pg* : le Périgord est en Gascogne): *[bp* **&** ***non*** *pg* **&** *bg]* ⇒ ***F***. Or cette nouvelle règle ne s'applique pas car *bp* est fausse. Le statut de *bg* devient inconnu, de même que celui de l'incohérence.

Nous retiendrons donc qu'une explication conversationnelle consiste dans l'explicitation d'une clause saturée ou dans son invalidation (directe ou indirecte)(3). Nous allons voir les conséquences de ce constat pour l'interaction avec les systèmes à base de connaissances.

---

[2] Ce refus de règle équivaut en fait à une invalidation du premier terme de la clause plus précise : *[A∩B=∅* **&** *X∈A* **&** *X∈B]* ⇒ ***F*** où X est Cyrano, A est la Dordogne et B la Gascogne.

[3] On peut considérer que certaines explications consistent dans l'explicitation ou l'invalidation non pas d'une incohérence, mais d'une improbabilité ou d'une (in)désirabilité (voir chap. I.1). Il semble toutefois que le besoin d'explication passe, même dans ces cas-là, par un étonnement, c.à.d. par un changement de modalité dans la relation improbable ou (in)désirable. Ce point est de toutes façons tout à fait marginal pour ce qui nous occupe ici, et n'a aucune conséquence pratique sur ce qui est exposé dans ce chapitre.

4. L'importance de la pertinence des explications pour les SBC

4.1. Les explications conversationnelles sont fortement contraintes

Nous avons montré dans le chapitre I.1 à quel point les conversations humaines étaient contraintes d'un point de vue logique. Parmi ces conversations, bon nombre sont organisées autour de la perception d'une incohérence. Dans ce cas, les interlocuteurs ne tolèrent que quatre types de répliques :

- des "co-étonnements", qui manifestent la compréhension de l'incohérence
- des clarifications du contexte
- des invalidations, directes ou indirectes, de l'incohérence
- des accords marquant l'acceptation d'une invalidation

Cet aspect contraint des interactions humaines est fondamental lorsqu'il s'agit d'amener un système artificiel à interagir sur des sujets nécessitant une relation langagière. Pour bien montrer à quel point les humains sont exigeants en matière d'explication, nous reproduisons ici un extrait que nous avons déjà donné dans le chapitre I.1 :

[ex_Goffman] (tiré de [Tannen 1984])

contexte: A, B et C parlent de sociologie, et il apparaît que B a une bonne connaissance des livres du sociologue Erving Goffman. A et C sont surpris, dans la mesure où ils pensaient que cet auteur n'était connu que des spécialistes.

A1- But anyway. ... How do you happen to know his stuff?
B1- Cause I read it.
C1- What do you do?
A2- [??] *are you in ... sociology or anything?*
B2- Yeah I read a little bit of it. [pronounced reed]
A3- Hm?
B3- I read a little bit of it. [pronounced red]
A4- I mean were you... uh studying sociology?
B4- No.
A5- You just heard about it, huh?
B5- Yeah. No. I heard about it from a friend who was a sociologist, and he said read this book, it's a good book and I read that book 'n...
A6- I had never heard about him before I started studying linguistics.
B6- Really?
A7- Yeah.

Cet extrait est particulièrement intéressant, car il montre à quel point une "explication" qui n'agit pas comme une invalidation peut être insatisfaisante. A et B sont étonnés que C ait une si bonne connaissance de l'oeuvre de E.Goffman alors que rien n'indique qu'il ait fait des études de sociologie. A et B détectent une incohérence, qui peut s'exprimer ainsi :

[connaît(X, livres_de_Goffman) & non sociologue (X)] $\Rightarrow$ F

Ils posent donc leur question A1. Mais, fait remarquable et pourtant prévisible dans notre modèle, la justification donnée à trois reprises par B (B1, B2, B3) ne les satisfait absolument pas. Effectivement, l'effet logique des répliques de B n'est pas d'invalider l'incohérence précédente :

lu(B**,** livres_de_Goffman) $\Rightarrow$ *connaît*(B, livres_de_Goffman)

On observe alors que A et C vont en quelque sorte *exiger* une invalidation acceptable de ce qui a provoqué leur étonnement, en posant leurs questions C1, A2, A4, A5. Ils obtiendront ce qu'ils cherchent, car B5 agit bien comme une invalidation indirecte. En effet, B5 invalide une version modifiée du paradoxe, que A est supposée accepter :

[*connaît* (**X**, livres_de_Goffman) **& non** *sociologue* (**X**)
& non *recommande*(**Y**, **X**, livres_de_Goffman)] ⇒ **F**

Noter que l'information apportée par B5 permet de bloquer l'apparition d'une nouvelle incohérence qui résulterait de l'application de ce contexte non plus à B, mais à l'ami de B (noté D) :

recommande(D, B, livres_de_Goffman) **&** *sociologue*(D)

Les explications conversationnelles apparaissent ainsi comme très fortement contraintes. On peut considérer que ces contraintes modélisent la *pertinence logique* de l'explication *q* :

Une explication n'est conversationnellement pertinente que si elle explicite ou invalide une clause saturée du contexte

Rappelons qu'une clause saturée est une clause négative dont tous les termes se retrouvent vrais. S'il s'agit d'une clause paradoxale (comme dans les exemples cités), l'explication est relative à une incohérence (voir note 3 p. 230).

4.2. *Génération d'explication dirigée par l'expertise* vs. *dirigée par les problèmes*

Nous défendons l'idée que les explications fournies par les systèmes à bases de connaissances doivent être calculées comme des invalidations ou des explicitations de contextes problématiques. Dans l'exemple d'interaction avec XPLAIN donné p. 226, le système donne une explication conversationnellement pertinente dans la mesure où il explicite un contexte indésirable (risque de fibrillation ventriculaire).

La situation est plus délicate lorsque la requête de l'usager est provoquée par un problème que ce dernier a rencontré. Nous suggérons fortement le fait que pour être pertinent, le système doit absolument fournir une information qui va invalider le caractère problématique de la situation perçue par l'usager. Or cela est rarement le cas.

En effet, la plupart des systèmes calculent les explications en fonction de leur propre perception de la situation et d'une éventuelle mesure des connaissances de l'usager. Ils ne passent pas par une phase d'identification du problème de l'usager. En d'autres termes, ils fonctionnent suivant un principe de **génération d'explications dirigée par l'expertise**, alors que nous suggérons une **génération d'explication dirigée par les problèmes**.

Par exemple, une architecture comme ESMERALDA [Lemaire 1992], dans laquelle tout est organisé pour adapter le niveau d'expertise de la machine à celui de son utilisateur, ne prévoit pourtant pas d'analyser le problème qui motive la requête de l'usager. L'extrait suivant est relatif au jeu de bridge :

ESM : . . . Declare one spade
User : why ?
ESM : Declare major first because fifthminor
User : what ?
ESM : Fifthminor is a five-card suit in minor. Let me resume. . . Just a definition. Fourthmajor is, for example, d, v, 8, 7 in spadesuit. Declare major first because fifthminor, fourhmajor biddable at level 1 and less than 11 pts.

User : why announce one spade ?
ESM : Just a definition. Strongopening is an opening with at least two honnors. Declare one diamond implies miss spade contract because the opene r may not have a strongopening with four spades.

Nous observons ici qu'ESMERALDA produit finalement deux explications, l'une causale, l'autre qualifiée de négative, entre lesquelles elle avait hésité dans un premier temps pour finalement choisir la première. Ce choix entre les explications est effectué en fonction de contraintes de temps de calcul, de connaissances prérequises, etc., sans que soit entreprise une analyse de la motivation de la requête.

Le fait que cette stratégie soit dirigée par la seule expertise n'a sans doute pas de conséquence rédhibitoire lorsqu'il s'agit de donner une définition à un utilisateur novice. De plus, dans certaines situations où le système effectue des actions, l'explicitation des motivations sous-jacentes peut, avec une bonne probabilité, résoudre le problème détecté par l'usager et donc constituer une explication pertinente.

Nous voulons toutefois insister sur l'insuffisance de tels principes, qui sont adoptés dans de nombreux modules d'explication. La génération d'explications dirigée par l'expertise ne peut, selon nous, conduire à des explications pertinentes que par chance. Si l'utilisateur se comporte comme il le fait dans les interactions humaines, il va solliciter le système chaque fois que la situation, telle qu'elle résulte de ses propres actions ou de celles du système, lui semblera incohérente. Quel que soit son niveau d'expertise, l'utilisateur a des *attentes*, et chaque fois que l'une d'entre elles est contredite, il s'attend à ce que le système lui fournisse une explication qui invalide logiquement son attente.

Dans le cadre de la recherche sur l'explication que nous menons en collaboration dans le projet GENE[4], nous souhaitons parvenir à la spécification d'un système capable de fournir des explications adaptées à son utilisateur. Nous avons donc proposé que toute tentative d'explication de la part du système débute par une analyse logique de la requête, de manière à ce que le caractère problématique de la situation perçue par l'utilisateur soit correctement identifié. Cette identification nécessite généralement une interaction dialogique, comme on peut l'observer avec des experts humains. L'extrait suivant reproduit une interaction entre un lecteur (L) et un bibliothécaire (R) dans un service de prêt de livres :

(extrait de [Polity et al. 1990], dialogue 24 p. 43)

L1: Excusez-moi, est-ce que vous avez passé tous les livres que j'ai rendus ?
R1: Oui, pourquoi, qu'est-ce qui s'est passé ?
L2: On me dit que j'ai trop de documents.
R2: Donnez-moi votre carte, je vais vous dire ce qu'il vous reste.
L3: Merci, je sais qu'il m'en reste deux.
R3: Ha, je ne vois rien ; c'est probablement parce que vous n'avez pas réglé votre cotisation à l'entrée.

Le phénomène étonnant, pour le lecteur, vient de ce qu'on ne le laisse pas sortir avec les ouvrages qu'il vient d'emprunter. Il est pourtant certain de ne pas avoir dépassé le seuil d'emprunt. L'explication donnée en R3 constitue une invalidation indirecte :

contexte de L1

[non trop_de_livres **&** on_me_bloque] ⇒ **F**

on_me_bloque

restitution_enregistrée ⇒ **non** trop_de_livres

[4] GENE : Gestionnaire d'Explications NEgociées, est un groupe de recherche du PRC-IA réunissant six chercheurs du CNRS et des Universités.

contexte de R3

[non trop_de_livres **&** on_me_bloque **&** cotisation_à_jour] ⟹ **F**

L'expert humain, R, est parvenu à résoudre le problème de L parce qu'il a compris que ce problème existait, et qu'il l'a identifié comme une incohérence. Un système qui n'aurait pas fait l'hypothèse d'une situation problématique se serait certainement contenté de répondre "oui" en R1, ce qui n'aurait pas beaucoup aidé le lecteur L. L'extrait suivant, qui est remarquable en raison de la similarité des situations, montre à quel point le calcul de l'explication dépend de l'identification du problème, et ne saurait être calculé *a priori* selon une méthode "dirigée par l'expertise" :

(extrait de [Polity et al. 1990], dialogue 21 p. 43)

L1: Est-ce que vous avez passé tous les livres ?
R1: Oui, tout est passé, qu'est-ce qu'il y a ?
L2: Je ne peux pas passer, parce que j'ai trop de livres; pourtant j'ai regardé à la maison, et...
R2: Vous avez votre carte ?
L3: Je vais la chercher
R3: Comme livre ancien, il y a H. de Montrouge, et "Vous n'avez pas vu J. Duflo", ... y a aussi Pouchkine
[. . .]
L9: Je vais vérifier aussi; j'ai dû ne pas le rendre, je vérifierai.
R9: D'accord, ça doit être pour ça que ça coince.
L10: OK.

Cette fois, l'explication est une invalidation directe de l'étonnement. Le lecteur a effectivement emprunté trop de livres.

4.3. La détection des requêtes problématiques par un SBC

Si l'on souhaite permettre à un SBC de situer ses explications par rapport au problème qui a motivé la requête de l'usager, on doit envisager certaines répercussions sur la conception du système et sur son utilisation, car l'étonnement doit être partageable. Le système doit disposer pour cela de connaissances explicites dépassant celles qui lui sont strictement nécessaires pour conduire ses inférences (*e.g.* règles de cohérence et règles stratégiques). On peut même aller plus loin et ne pas discerner ces trois types de connaissances (inférentielle, cohérence, stratégique) dans le module d'explication, puisque toutes peuvent mener à des étonnements.

Notre système SAVANT3 (voir chap. III.2) peut produire des explications basées sur la détection d'une incohérence. Il s'agit toutefois d'un SBC un peu particulier. Par exemple, il ne comporte aucune distinction entre les règles d'inférence et les contraintes de cohérence (tout se passe comme si SAVANT3 ne comportait que des "règles de production de l'incohérence" (comme celle qui sont introduites dans le système COVADIS tel qu'il est décrit dans [Ayel & Rousset 1990]). SAVANT3 ne comporte pas non plus, comme on l'a vu, de règles stratégiques. Mais si l'on voulait en intégrer, il faudrait le faire explicitement, comme l'a montré Clancey [1987] en créant GUIDON, de manière à pouvoir s'étonner de choix stratégiques effectués par l'étudiant.

SAVANT3 repose sur le principe du partage de l'étonnement. Mais ce partage n'est possible avec un SBC que si le système se met à la portée de son utilisateur. L'étude logique de l'explication conversationnelle permet ainsi de confirmer dans le détail une idée intuitive souvent rappelée : une communication conceptuelle basée sur l'explication n'est possible qu'entre deux entités disposant de connaissances déjà très proches. Dans le cas de SAVANT3, l'étudiant doit absolument être conscient de la contradiction par rapport à laquelle le système se situe. Cela exige en particulier que le système manipule les mêmes concepts que l'étudiant (ce qui exclut qu'il y ait des termes obscurs pour l'étudiant), et que la plupart des règles soient partagées.

Dans le cas des SBC traditionnels, cette exigence a peu de chances d'être satisfaite, du fait de la distance entre règle d'inférence et règle logique. Certaines règles d'un SBC ont une justification heuristique ou statistique. Elles traduisent l'expertise et ont peu de chances d'être connues de l'utilisateur. De plus, l'explication d'un étonnement nécessite que toutes ses prémisses soient explicitables. Or il est fréquent que des prémisses "toujours vraies" soient omises des règles d'inférence.

On peut imaginer toutefois de nombreuses situations dans lesquelles le type d'explication que nous prônons soit accessible à un SBC. Ce seront nécessairement des situations où les connaissances du système et celles de son utilisateur seront extrêmement proches: dialogue expert-SBC, système d'aide contextuelle avec modèle des connaissances de l'utilisateur, etc.

5. L'explication comme une critique argumentée

5.1. Critiquer plutôt que résoudre

Dans l'explication telle que nous venons de l'évoquer, le système à base de connaissance est généralement censé détenir la solution du problème de l'utilisateur. Si l'on conçoit l'explication non comme une solution, mais comme une critique, on obtient une conception beaucoup plus souple du rôle du système à base de connaissance.

Une critique n'est pas un jugement de validité définitif, mais une sorte de marque de désaccord potentiel. Les critiques portent sur les actions (ou les croyances) de l'utilisateur, et peuvent être déclenchées, par exemple, par la détection d'une situation anormale (par ex. une situation incohérente ou indésirable au cours de la résolution de problème). C'est la situation de critique *active*, dans laquelle le critique regarde comme on dit "par-dessus l'épaule" de l'utilisateur, par opposition à la critique *passive* déclenchée sur l'initiative de ce dernier.

De tels systèmes critiques existent, par exemple les systèmes ACTIVIST, LISP-CRITIC, FRAMER, etc. développés à Boulder par l'équipe de G. Fischer [Fischer et al. 1991].

ACTIVIST est un système d'aide actif pour éditeur de texte. Il observe les actions de l'utilisateur de l'éditeur, et en infère ses buts. Dès qu'il a pu observer trois actions qu'il peut interpréter comme l'exécution répétée d'une stratégie sous-optimale, ACTIVIST informe l'utilisateur de l'existence d'une meilleure procédure pour atteindre le même but.

LISP-CRITIC est un autre critique, conçu pour l'aide à la programmation en langage Lisp. Il suggère des transformations du code écrit par l'utilisateur qui rendent son programme plus lisible, plus concis ou plus efficace. Par exemple, il peut suggérer de remplacer une expression conditionnelle à choix multiples : *(cond (C a) (t b))* par une expression plus simple *(if C a b)* dans un certain contexte.

Nous avons envisagé d'utiliser notre système SAVANT3 comme un critique pour aider à l'acquisition de savoir-faire complexes. Cela n'allait pas de soi, puisque SAVANT3 est avant tout conçu pour l'enseignement des connaissances conceptuelles, et non pour celui des savoir-faire. Nous avons cependant mentionné l'importance des concepts dans l'acquisition des savoir-faire complexes (chap. III.1).

Notre approche consiste donc à utiliser SAVANT3 pour apporter une critique de type conceptuel à l'étudiant. Nous allons en voir un exemple sous la forme d'une simulation argumentée pour l'apprentissage de Prolog. Mais notre démarche diffère de celle des autres systèmes critiques sur un point qui nous semble important. Dans son principe, la simulation argumentée que nous présentons ici ne consiste pas à fournir un simple diagnostic. Il s'agit, à partir de ce que SAVANT3 considère comme une anomalie de structure dans le programme de l'étudiant, de susciter un *échange argumentatif* avec ce dernier. On rejoint ainsi l'idée de base de SAVANT3 qui utilise une "conversation" pour tester la cohérence des connaissances de l'étudiant, et non une simple succession d'échanges question-feedback [Dessalles 1993c]. Cette approche semble particulièrement indiquée dans le cas de l'apprentissage d'un langage de programmation, et ce pour plusieurs raisons :

- La possibilité de faire tourner le programme écrit par l'étudiant permet à celui-ci de constater des exécutions anormales, ce qui est censé le motiver pour solliciter la critique de SAVANT3[5].
- Les critiques que l'on peut formuler sur un programme donné sont rarement définitives. L'étudiant peut avoir une motivation, non connue du critique, pour écrire par exemple une clause correcte mais très inefficace. L'échange argumentatif lui permet d'imposer son point de vue.
- Le diagnostic immédiat permet rarement de mettre la faute conceptuelle en évidence. C'est souvent lorsque l'élève se défend en donnant ses raisons que le dialogue remonte jusqu'aux présupposés erronés.

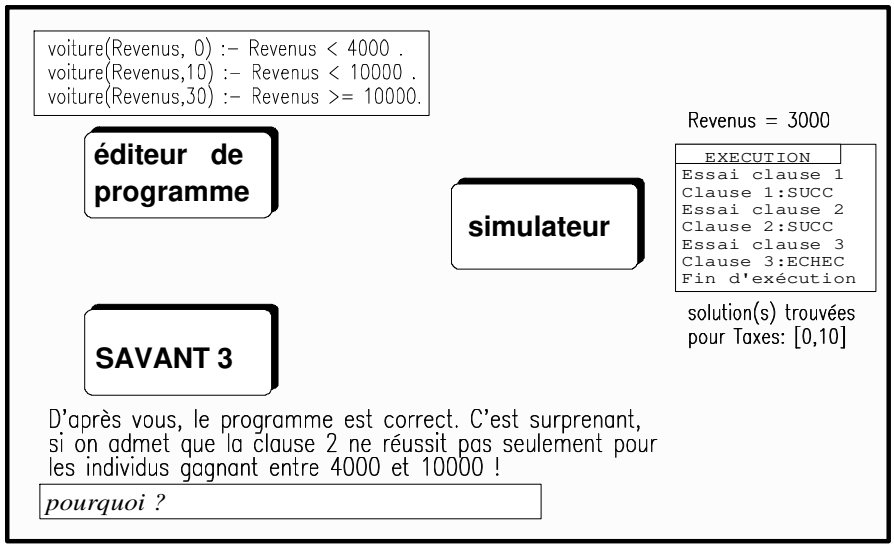
5.2. *SAVANT3 dans le rôle d'un critique : une simulation argumentée pour l'apprentissage de Prolog*[6]

Dans la maquette que nous avons mise au point, on demande à l'étudiant d'écrire un petit programme Prolog qui calcule la relation *voiture(**X**,**Y**)* suivante :

Pour **X** inférieur à 4000, **Y** doit être nul.
Pour **X** entre 4000 et 10000, **Y** doit être égal à 10.
Pour **X** supérieur à 10000, **Y** doit valoir 30.

Ce programme est supposé calculer la taxe (**Y**) qu'un individu doit payer sur sa voiture d'après le montant de ses revenus (**X**). Mais nous ne laissons pas à l'étudiant la possibilité d'écrire un programme quelconque. Nous lui fournissons un éditeur très limité qui impose des contraintes sur la formes des clauses qu'il peut écrire. La figure ci-dessous donne un exemple de programme.

A chaque instant l'étudiant peut exécuter son programme : il fournit la valeur de ***Revenus*** dans *voiture(**Revenus**, **Taxes**)*, puis l'interpréteur Prolog cherche toutes les solutions possibles pour ***Taxes*** pendant qu'une fenêtre de trace montre les clauses appelées et leur échec éventuel (voir figure).



Dans la maquette que nous avons mise au point, on demande à l'étudiant d'écrire un petit programme Prolog qui calcule la relation *voiture(**X**,**Y**)* suivante :

---

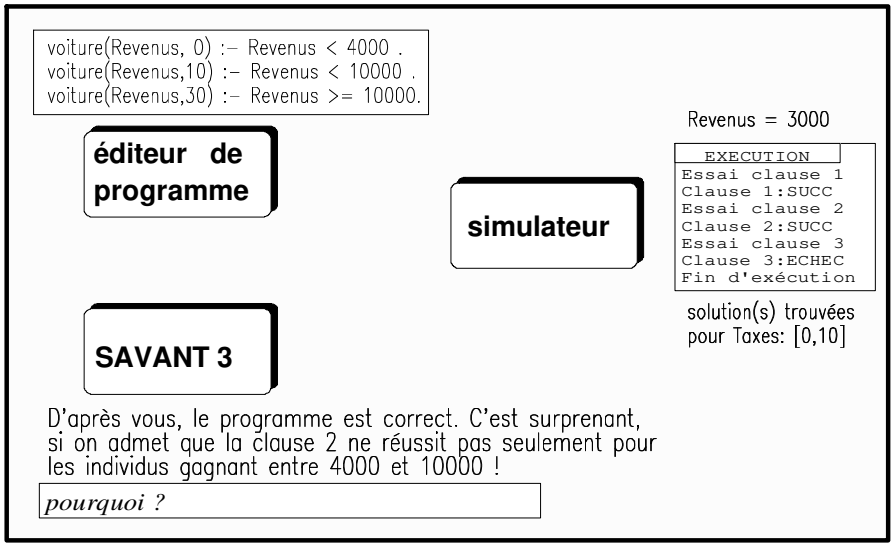
[5] On peut imaginer que SAVANT3 intervienne automatiquement en cas d'anomalie (critique active), mais nous n'avons pas encore implanté cette possibilité.

[6] Ce travail a été réalisé en collaboration avec Pascal Meyers, de la Katholieke Universiteit Leuven, et publié dans [Dessalles & Meyers 1993].

Pour **X** inférieur à 4000, **Y** doit être nul.
Pour **X** entre 4000 et 10000, **Y** doit être égal à 10.
Pour **X** supérieur à 10000, **Y** doit valoir 30.

Ce programme est supposé calculer la taxe (**Y**) qu'un individu doit payer sur sa voiture d'après le montant de ses revenus (**X**). Mais nous ne laissons pas à l'étudiant la possibilité d'écrire un programme quelconque. Nous lui fournissons un éditeur très limité qui impose des contraintes sur la formes des clauses qu'il peut écrire. La figure ci-dessous donne un exemple de programme.

A chaque instant l'étudiant peut exécuter son programme : il fournit la valeur de ***Revenus*** dans *voiture(**Revenus**, **Taxes**)*, puis l'interpréteur Prolog cherche toutes les solutions possibles pour ***Taxes*** pendant qu'une fenêtre de trace montre les clauses appelées et leur échec éventuel (voir figure).



Cet exercice est utilisé pour enseigner à l'étudiant certains savoir-faire nécessaires pour l'écriture et la compréhension de programmes Prolog. L'objectif est d'attirer son attention sur la justesse et l'efficacité de son programme.

L'interprète Prolog ne peut pas mettre clairement en évidence le manque d'efficacité du programme. Il peut simplement montrer que le programme a un comportement anormal, mais il ne peut pas relier ceci à une faute conceptuelle éventuelle sur le fait, par exemple, qu'il y a backtracking et que le programme peut donner plusieurs solutions, ou encore sur la signification des cuts, etc. C'est en cela que l'intervention de SAVANT3 peut être utile.

A tout moment l'étudiant peut soumettre son programme à SAVANT3, qui s'efforce de découvrir une faille (*i.e.* de saturer l'une de ses règles) :

extrait 2

```
voiture(Revenus,0) :- Revenus < 4000, !.
voiture(Revenus,10):- Revenus >= 4000, Revenus < 10000,
                      !.
voiture(Revenus,30).
```

S3- D'après ce que j'ai compris, la clause 2 contient [Revenus >= 4000]. Mais peut-être pensez-vous que le programme est le plus efficace possible ?
Et.- Pourquoi ?
S3- Il y a tout de même un problème... sauf si vous considérez que la clause 2 est appelée pour les individus gagnant moins de 4000
Et.- oui

S3- D'après vous, la clause 2 est appelée pour les individus gagnant moins de 4000. C'est surprenant si l'on admet que la clause 1 réussit pour les individus gagnant moins de 4000 !
Et.- pourquoi est-ce surprenant ?
S3- Il y a tout de même un problème... D'après ce que j'ai compris :
- *la clause 2 est appelée pour les individus gagnant moins de 4000*
- *la clause 1 réussit pour les individus gagnant moins de 4000*
- *il y a un cut dans la clause 1*

Que proposez-vous comme solution ?
Et.- [la clause 2 n'est pas appelée pour les individus gagnant moins de 4000]
S3- *Mais attendez... D'après ce que vous avez dit :*
- *la clause 2 contient [Revenus >= 4000]*
- *la clause 2 n'est pas appelée pour les individus gagnant moins de 4000*

Je m'avoue surpris !
Et.- j'enlève [Revenus >= 4000] de la clause 2
S3- Ah bon!... Ok...

Nous avons rencontré certaines difficultés techniques pour la réalisation de cette simulation argumentée. La principale tient à la constitution de la base de connaissances qui permet à SAVANT3 d'émettre ses critiques. Alors que 15 ou 20 règles suffisent habituellement sur des sujets purement conceptuels (cf. chap. III.2), il en faut presque dix fois plus pour gérer la critique de ce simple exercice de Prolog ! Nous perdons ainsi le principal intérêt de SAVANT3 qui est le faible coût de conception des contenus. Nous travaillons actuellement à une définition plus abstraite de la connaissance relative à la simulation. Une telle connaissance serait ensuite "compilée" au moment de l'interaction, en fonction de la situation créée par l'étudiant.

Le diagnostic d'une faute conceptuelle est difficile, voire impossible à partir d'une performance unique [Wenger 1987]. Le pari qui est à la base de notre approche est qu'un tel diagnostic n'est pas nécessaire pour que le système commence à argumenter. Le système détecte une anomalie. Si cette anomalie est la conséquence d'une faute conceptuelle, une argumentation bien menée doit permettre à l'élève de prendre conscience de l'origine de la faute.

6. Conclusion : le respect des contraintes cognitives de l'utilisateur

Le type de communication qui s'instaure dans les interactions humaines semble dépendre de celui qui contrôle les conditions de la communication. Plus précisément, lorsque la communication est contrôlée par celui qui possède la connaissance, comme c'est le cas par exemple dans un exposé magistral ou dans le cadre un peu formel d'une réunion de travail, nous avons pu montrer que le type d'explication fourni était organisé en fonction des contraintes cognitives de l'émetteur [Dessalles 1992b]. En revanche, dans les conversations spontanées, la communication semble être plutôt contrôlée par le récepteur, c'est-à-dire par celui qui ne possède pas l'élément d'explication. Or dans ce deuxième cas, les explications émises semblent systématiquement dépendre d'une situation problématique correctement perçue aussi bien par l'émetteur que par le récepteur de l'explication.

Cette constatation nous semble fondamentale, et nous permet de penser que l'exigence que toute explication réponde à une relation problématique est l'indice d'une contrainte cognitive propre à celui qui reçoit l'explication (cf. chap. II.3 et II.4).

Nous suggérons ainsi le fait qu'une meilleure acceptabilité des systèmes à base de connaissances peut être obtenue si les contraintes cognitives de l'utilisateur sont respectées, autrement dit si toutes les explications du système agissent correctement comme des invalidations (ou des explicitations) de relations problématiques perçues par l'utilisateur. Si l'on accepte ce point de vue, alors les systèmes à base de connaissances doivent être conçus de manière à pouvoir fournir des *explications de type conversationnel*.

L'explication de type conversationnel n'est pas *dialogique* par nature ! On peut imaginer que l'explication joue son rôle d'invalidation (ou d'explicitation) du premier coup. Mais ces cas sont rares dans la pratique. L'identification de la relation problématique exige souvent elle-même plusieurs échanges (cf. plus haut les extraits du bibliothécaire). Par ailleurs, une explication qui invalide la relation problématique initiale risque de créer un nouveau problème, et donc un nouveau besoin d'explication. C'est ce que l'on observe couramment dans les conversations. L'explication de type conversationnel est donc dialogique par nécessité pratique !

L'explication de type conversationnel n'exige pas, contrairement à une vision traditionnelle, que le système soit omniscient. L'intérêt de considérer le SBC comme un *critique* est justement de permettre au système de ne fournir que les explications qu'il peut fournir, c.à.d. celles qu'il peut raccrocher à une situation problématique. L'intervention du critique suppose que celui-ci a été capable de déceler une anomalie, et l'explication qu'il produira sera bien de type conversationnel : explicitation de l'anomalie, puis élément de solution permettant de supprimer l'anomalie. L'apport de la vision conversationnelle, par rapport aux systèmes critiques habituels, est de concevoir la critique sous forme *dialogique*.

Notre façon de concevoir l'explication dans les SBC comme devant se conformer aux contraintes de la conversation nous permet de proposer une convergence entre la vision *explicative* et la vision *critique* du rôle du SBC. Il reste bien entendu de nombreuses recherches à effectuer avant de pouvoir démontrer la faisabilité technique de l'explication conversationnelle pour des SBC réels, puis pour montrer que l'explication conversationnelle entraîne une meilleure acceptabilité des SBC.