
Aide interactive pour un outil de conception

Karine Duprez

*INRIA Rhône-Alpes, projet Sherpa
ZIRST, 655 avenue de l'Europe
F-38330 Montbonnot Saint Martin, France
duprez@inrialpes.fr*

Résumé : *Dans le domaine de la conception assistée par ordinateur, un outil est initialement dédié à un domaine restreint et s'adresse aux experts, aux concepteurs professionnels. L'introduction d'un système d'aide vise à rendre l'outil plus accessible : pour des utilisateurs occasionnels ou pour une prise en main par des néophytes. Tous ces utilisateurs partagent une connaissance du domaine d'application, mais demandent une aide pour une meilleure utilisation de l'outil. Il s'agit alors de mettre en oeuvre des facilités sans nuire à la créativité de l'utilisateur et sans contraindre ses possibilités. La proposition consiste à éduquer l'utilisateur, en l'impliquant dans sa recherche d'informations pour l'inviter à se passer rapidement de l'aide. Le système d'aide devient un système d'aide à la décision où les suggestions sont expliquées pour laisser l'utilisateur prendre sa décision en toute connaissance de cause.*

Mots clés : *Aide contextuelle, logique de fonctionnement et d'utilisation, outil de conception, aide à la décision, représentation de connaissances*

1- Introduction

De nombreux logiciels assistent les différentes étapes du processus de conception. Nous nous limitons ici au domaine de la conception de circuits en nous attachant à l'étape de la simulation. La diversité des outils de simulation présents sur le marché demande aux concepteurs de changer facilement d'environnement de travail. Tous les projets ne se déroulent pas avec les mêmes outils, et ces derniers ne sont pas toujours compatibles. Il est courant que seule une faible partie des capacités d'un logiciel soit utilisée : les autres aspects ne sont pas indispensables pour le travail spécifique. Ainsi la diversité, et la richesse propre des outils sont les facteurs qui limitent l'expertise des utilisateurs. Cependant les concepteurs ont une connaissance experte, restreinte à leur emploi spécifique de l'outil.

Nous verrons tout d'abord les particularités d'un outil de conception à prendre en compte pour préciser les besoins d'aide des utilisateurs potentiels. Nous discuterons ensuite les approches envisagées pour aider les concepteurs et nous exposerons les limites d'une aide qui ne préserve pas le statut d'expert d'un concepteur. Nous proposerons enfin notre approche d'aide adaptée au contexte et le travail réalisé dans une phase de maquettage.

2- Les outils de conception

Le terme outil recouvre le logiciel de conception, qui peut lui-même regrouper plusieurs fonctionnalités. Notre travail est illustré dans le domaine de la simulation de circuits électroniques qui fait intervenir les principales notions attachées à un outil de simulation :

- Le domaine d'application, qui sera l'électronique.
- Les utilisateurs d'un logiciel professionnel de simulation, qui partagent les connaissances de base du domaine en tant que **concepteurs**. Cependant tous les concepteurs n'ont pas la même expérience d'un simulateur donné. Les utilisateurs occasionnels, contrairement aux experts en simulation, se forment rarement à l'ensemble de l'outil, et limitent leur découverte aux fonctionnalités nécessaires pour accomplir leur travail. Néanmoins, ils ont une connaissance empirique sur les fonctionnalités employées et ils se comportent comme des experts sur ces points connus. Les néophytes en simulation ont pour seul a priori les outils précédemment utilisés pour réaliser un travail équivalent.
- La description de modèles pour appréhender le monde réel. Le circuit est représenté par sa topologie et la modélisation du comportement de ses composants. La qualité de la simulation dépend de la qualité de la modélisation. Or un modèle approxime le monde réel en valorisant certaines facettes que le concepteur juge importantes d'étudier. Il n'est pas possible de concevoir des modèles prenant en compte toutes les caractéristiques physiques d'un composant électronique sur la plage de fonctionnement du composant. La qualité du modèle n'est pas intrinsèque, mais dépend de la précision attendue pour un type d'analyse.

Le simulateur peut être mis à contribution au cours de plusieurs étapes de la conception. Ainsi le concepteur peut avoir besoin de vérifier une spécification en utilisant une modélisation fonctionnelle (avec des modèles comportementaux). Une utilisation plus classique consiste à prendre le simulateur pour un oscilloscope virtuel pour visualiser au cours du temps la tension et/ou le courant dans le circuit. Les analyses du circuit en régime fréquentiel et en régime permanent sont aussi disponibles. Interviennent ensuite des études du signal (bruit ou transformée de Fourier).

Les outils de conception soulèvent des problèmes relatifs à la liberté accordée à l'utilisateur. La simulation quant à elle doit inciter l'esprit critique de l'utilisateur. Nous précisons notre point de vue sur les trois aspects suivants.

2.1- La créativité et l'initiative

La conception met en jeu la créativité des concepteurs : il s'agit d'utiliser l'expérience pour résoudre un nouveau problème. Pour travailler, un concepteur n'admet pas d'être contraint par l'outil de conception dans ses choix, et encore moins de se voir imposer une solution. Autant les contraintes de la spécification, donnée par un client, constituent l'objectif à respecter, autant les moyens pour atteindre cet objectif relèvent du choix du concepteur. L'outil de conception ne doit pas entraver cette liberté, ni brider les initiatives du concepteur.

2.2- Le besoin de contrôle

Il est évident que le néophyte en simulation ne refusera pas des directives pour le guider dans la prise en main de l'outil. Cependant, dès que le concepteur a un peu découvert l'outil, sa demande est de mieux comprendre « ce qui se passe au sein du simulateur », et il souhaite savoir comment contrôler et influencer la simulation.

Le concepteur préfère parfois opérer manuellement pour avoir un contrôle complet. Ainsi quand l'état d'équilibre du circuit mis sous tension n'est pas obtenu par le simulateur, il est parfois intéressant d'ajouter des résistances au voisinage des transistors. Il existe des commandes qui procèdent de manière automatique sur tous les transistors présents dans le circuit. Néanmoins, des concepteurs expérimentés, qui connaissent les commandes automatiques, vont ajouter manuellement des résistances, en choisissant l'emplacement et la valeur de ces composants. La raison invoquée est « d'être plus électrique » (plus proche d'une modification physique du circuit), et de savoir exactement ce qui est modifié dans la nouvelle tentative. La demande de contrôle sur l'outil est forte car elle permet au concepteur d'être maître des modifications et d'appliquer ses connaissances individuelles (intuitions, idées, astuces, expériences...) à son problème.

2.3- L'attitude critique face à un simulateur

Comme indiqué précédemment, la simulation peut intervenir à différentes étapes de la conception, mais toujours pour vérifier, valider un choix. Le concepteur doit garder à l'esprit que la simulation n'est jamais qu'un reflet du processus physique simulé, qui dépend de la définition du processus, de la modélisation, des analyses et des algorithmes employés. Si la simulation ne procure pas le résultat escompté, la difficulté est d'identifier le problème, pour savoir où les modifications sont nécessaires, en sachant qu'il n'y a pas une unique solution. La difficulté supplémentaire apportée par la simulation est de préserver l'esprit critique du concepteur, car ce dernier doit remettre en cause les résultats s'ils diffèrent de son attente, ou s'ils sont inexplicables. L'attitude critique est d'autant plus importante que seul le concepteur fixe les tolérances des algorithmes, et juge de la qualité de la simulation. La qualité d'une simulation dépend de la précision et de la rapidité, deux facteurs antagonistes pour lesquels il faut trouver un compromis.

3- L'aide à l'utilisation des outils de conception

Notre travail concerne **l'aide à l'utilisation** d'un outil informatique dédié à la simulation. Il est important de ne pas le confondre avec une aide à la conception, qui consisterait dans notre exemple à proposer des structures de circuits pour répondre à une spécification client. Le but serait alors de répondre à des questions « Comment optimiser un circuit vis-à-vis de la consommation d'énergie, ou vis-à-vis de la taille... ». Pour aider à la conception, au dessin du circuit, il faudrait alors manipuler et faire comprendre des notions de base comme le courant, les lois physiques de Kirchoff, expliquer ce qu'est chaque composant et quels sont les intérêts et les contraintes de son utilisation. Notre travail se limite aux connaissances concernant l'outil et ses capacités vis-à-vis du circuit étudié. Il est clair que les notions de base sont nécessaires pour appréhender un simulateur, mais tout concepteur connaît les bases de l'électronique.

L'aide à l'utilisation n'aborde pas les connaissances relatives à la plate-forme informatique. La manipulation de la souris ou la gestion du multi-fenêtrage dépend du système d'exploitation : cette manipulation sera renseignée dans l'aide de la plate-forme (Unix, système Macintosh, Windows...) accueillant le simulateur. Dans le cadre de notre application, la disponibilité de l'outil sous plusieurs plates-formes a permis de clairement poser les limites de l'aide souhaitée.

Après la description des besoins des utilisateurs, nous abordons les axes déjà étudiés pour apporter une aide. Nous discutons ensuite des possibilités pour satisfaire l'ensemble des utilisateurs et concluons par les choix retenus dans notre approche pour offrir un statut d'expert à l'utilisateur.

3.1- Les besoins des utilisateurs

Notre travail doit respecter les points revendiqués par les concepteurs interrogés : ne pas être bridés dans leurs initiatives, ne pas restreindre leur contrôle sur l'outil de conception. Ces revendications doivent être prises en compte pour les trois principales requêtes détaillées :

- Découvrir les fonctionnalités et les options de l'outil de manière interactive. L'outil de simulation est trop riche pour être exhaustivement connu, et l'utilisateur consulte rarement le manuel de référence. Il est donc nécessaire d'offrir une aide disponible à tout moment, pour expliquer « Comment une commande est activée? », « Quel est le rôle d'un paramètre? »...
- Connaître au moins une méthode pour réaliser un objectif. Cette requête correspond à « Comment faire pour...? » et offre à l'utilisateur un schéma directeur pour réaliser une étude Monte-Carlo par exemple.
- Apprendre à dépasser une situation d'échec quand le simulateur ne parvient pas à appliquer un algorithme sous les conditions fixées par l'utilisateur. Les causes peuvent être liées à une mauvaise description du circuit (par exemple une modélisation irréaliste, des composants pas connectés...), à une tolérance de convergence trop contraignante pour obtenir une solution numérique, à un algorithme inadapté. Il n'est pas simple d'identifier les causes d'erreur, ni d'y remédier car il n'existe pas une unique solution. Il est nécessaire de procéder par essais et échecs pour trouver une solution « qui marche », qui pourra ensuite être optimisée.

3.2- Les différents axes pour apporter une aide

Des aides pour l'utilisation d'un outil de conception existent. Certains simulateurs offrent un exemple de simulation, menée pas à pas pour un circuit donné (tutorial), ce qui permet de découvrir comment faire les analyses illustrées. D'autres proposent aussi une aide en ligne présentant le manuel de référence pour répondre à la requête sur les fonctionnalités. Malheureusement les aides apportées ressemblent beaucoup trop aux manuels. Elles ne tirent pas partie de l'interaction, et ne répondent pas aux besoins des concepteurs signalés précédemment. Quand le manuel de l'utilisateur est insuffisant pour remédier à un échec, l'aide est offerte par le support (hot-line), qui peut tenir compte du contexte de la simulation et de l'expérience des problèmes précédemment rencontrés.

3.2.1- Une aide contextuelle

La séparation des requêtes, entre une aide ponctuelle sur un paramètre et une aide guidée par les objectifs, est influencée par [Richard83] qui distingue la logique de fonctionnement et la logique d'utilisation. L'aide correspondant à la logique de fonctionnement est déjà formalisée dans le manuel de référence, mais il est intéressant d'offrir une aide contextuelle : signaler que le paramètre interrogé n'a aucune influence sur le travail courant car, par exemple, le paramètre concerne les circuits logiques alors que le circuit étudié est analogique.

Des travaux pour faire coopérer une aide contextuelle et une logique de fonctionnement sont proposés par [Tarby94]. Le contexte est déduit grâce à la logique d'utilisation en modélisant l'outil. Cette modélisation est basée sur les buts à atteindre avec leurs plans respectifs (enchaînement donné par une planification hiérarchique des tâches). Cette approche de la logique d'utilisation impose d'une part que l'utilisateur déclare son but, et d'autre part qu'il suive un des plans retenus dans la logique d'utilisation. Les connaissances de l'utilisateur sont alors vues comme un sous-ensemble des connaissances du système d'aide : l'utilisateur se sert de l'outil dans les limites prévues par le système d'aide, et il ne pourra pas envisager un déroulement de son travail qui sort d'un plan connu du système. Cette approche est intéressante pour les processus dont les déroulements sont exhaustivement connus.

Cependant elle est trop rigide pour un processus de conception, au cours duquel l'utilisateur demande à tester ses propres intuitions. Exploiter la démarche de l'utilisateur comme un filtre, en la comparant aux plans connus par le système pour identifier l'intention de l'utilisateur, présente aussi un risque d'oeillère. Un plan plus adapté risque d'être masqué par la démarche de l'utilisateur [Mais89], sans que celui-ci en soit averti. Il est vrai que le souci d'optimisation n'est pas nécessairement prioritaire, mais pourquoi ne pas laisser le choix à l'utilisateur, en présentant les solutions envisagées par le système d'aide, avec les avantages et les inconvénients de chaque solution? L'utilisateur pourra alors opter pour une solution « d'opérationnalisation », mais en sachant qu'il existe d'autres possibilités à envisager pour une éventuelle optimisation. De plus cette approche par l'explication permet de faire comprendre à l'utilisateur qu'il n'y a pas de solution optimale universelle car il existe au moins deux critères antagonistes : la précision et la vitesse de la simulation.

3.2.2- Une aide à la décision

L'aide sur les situations d'échecs est trop souvent du ressort du support téléphonique (hot-line). L'utilisateur décrit son problème à un opérateur, et ce dernier tente successivement des modifications adaptées à la situation, à défaut de pouvoir préciser la cause de l'échec. La procédure essai et échec, qui consiste à modifier les paramètres qui influencent la simulation et à relancer l'analyse, est donc utilisée par les utilisateurs experts pour rechercher une solution. L'ordonnancement des tentatives dépend de l'expérience individuelle de l'opérateur, des paramètres pressentis influents, et il est difficile d'extraire une connaissance consensuelle. Proposer un système qui automatiquement teste une succession de modifications va à l'encontre de la responsabilisation de l'utilisateur, qui contactera à nouveau le support sur un autre problème, parfois similaire.

Pour répondre aux requêtes utilisateurs en cas d'échec, il apparaît important d'expliquer pourquoi des causes sont retenues. Un système expert classique, qui offre une unique solution, n'est pas adapté à l'attente d'un concepteur. Le concepteur souhaite en effet savoir quelles possibilités ont été étudiées par le système pour évaluer la richesse des connaissances du système d'aide. Il souhaite connaître les motivations (avantages et inconvénients) de la proposition du système d'aide, pour juger de la pertinence des propositions et pour adapter en connaissance de cause une proposition du système d'aide à son cas particulier.

Implicitement, le statut d'expert est accordé au concepteur dans la prise de décision. Cette mise en valeur de la capacité de décision de l'utilisateur est aussi avancée dans les travaux de [Forslund95]. **Le système est alors un système d'aide à la décision**, où seul l'utilisateur prend les décisions, qui ne correspondent pas nécessairement aux suggestions du système d'aide. Le système d'aide n'a pas alors pour but de trouver une solution, mais d'apporter les informations manquantes à l'utilisateur au moment opportun afin de fixer les nouvelles connaissances par la construction d'une solution. Le système d'aide présente les avantages et inconvénients des suggestions dans le contexte présent. La liberté du concepteur est respectée, et les décisions sont prises en connaissance de cause.

3.3- Une aide adaptée à la diversité des utilisateurs

Les utilisateurs d'outil de CAO sont des concepteurs experts mais ils peuvent être distingués comme utilisateurs néophytes, occasionnels ou expérimentés. Cependant la distinction est toute relative tant l'expertise est liée à une fonctionnalité ou à un type d'analyse. Ainsi un utilisateur occasionnel peut connaître de manière experte quelques fonctionnalités, en ignorant des facilités de l'outil sollicitées pour d'autres aspects de la conception : il n'est ni expert, ni néophyte. Cet aspect est pris en compte par [Berthomé95], avec un système d'aide qui construit le profil de l'utilisateur pour chaque fonctionnalité, et l'adapte en cours

d'utilisation. Cette approche évite le classement grossier par niveau (néophyte, avancé, expert...). Ces remarques nous amènent à ne pas négliger les concepteurs expérimentés au profit des néophytes, et à ne pas s'appuyer sur un profil rigide d'utilisateur. L'attitude naturelle envers un débutant est de le décharger des choix et des réglages multiples, en automatisant au maximum le déroulement de son travail. Cependant, cette attitude n'est pas formatrice et répond aucunement au besoin de contrôler l'outil. Pour ne pas déboussoler ces utilisateurs qui ne connaissent pas la simulation, il est intéressant de suggérer une solution par défaut pour ne pas imposer à l'utilisateur de faire un choix.

Le profil de l'utilisateur est une base pour construire des explications compréhensibles par l'utilisateur. Depuis longtemps retenu en enseignement assisté par ordinateur, la « modélisation cognitive » de l'utilisateur semble intéressante pour une aide à l'utilisation [Senach87]. Il est communément admis [Cawsey92], [Chin89], d'envisager la modélisation des connaissances (identification des connaissances, niveau de compréhension, représentations mentales) pour comprendre la démarche suivie par l'utilisateur et pour expliquer à ce dernier une situation en des termes et des concepts qui lui sont familiers. Cependant, l'interaction nécessaire entre le système d'enseignement et l'utilisateur est trop importante. Il n'est pas souhaitable d'interrompre un concepteur professionnel en cours de travail, que ce soit pour une évaluation via des tests, ou pour confirmer des intentions. Il est indispensable de masquer l'aide lorsque le concepteur souhaite travailler sans interruption : suffisamment d'informations doivent donc être obtenues à l'insu de l'utilisateur pour que l'aide suive toute évolution du contexte de travail.

3.4- Une nouvelle approche

Il nous semble intéressant d'impliquer l'utilisateur dans sa recherche d'informations et d'explications, plutôt que de modéliser le profil de l'utilisateur. L'idée est de rendre l'utilisateur actif, de l'amener à une utilisation complète de l'outil. L'approche d'un **utilisateur actif** dans sa quête d'information **permet de contourner le problème d'évaluation de niveau**. La meilleure image du système l'aide que nous souhaitons proposer est aujourd'hui l'aide du système Macintosh 7.5, construite sur des couches d'informations. Chaque sujet est expliqué avec plusieurs niveaux de détails, et les notions mentionnées dans les explications peuvent aussi être consultées. Cette approche permet d'offrir une description certes laconique mais suffisante pour un utilisateur confirmé, et permet au néophyte de rechercher, pour une notion, plus d'informations. L'aide du système Macintosh surprend par sa convivialité et la mise en valeur de l'initiative de l'utilisateur : ce dernier décide des notions à vérifier, du niveau d'information nécessaire pour mener son travail. Bien sûr, l'image est incomplète car la notion de contexte n'est pas intégrée dans le système d'aide du Macintosh, et les explications sont consensuelles contrairement au problème rencontré pour la conception. Cette approche répond au souhait de se passer rapidement de l'aide, car la logique d'utilisation est poussée jusqu'à la mise en pratique immédiate. En résumé, le meilleur moyen d'apprendre à faire des choses est de les faire.

L'interactivité proposée entre le système d'aide et l'utilisateur correspond à l'implication de l'utilisateur dans ses choix. L'interaction est possible si le système d'aide et l'utilisateur sont actifs : l'un en étant contextuel, l'autre en étant le décideur. Nous récapitulons les choix retenus pour offrir une aide interactive et contextuelle.

- Accorder un statut d'expert à tout utilisateur, en l'incitant à rechercher les informations qui lui manquent en parcourant les couches d'explications.
- Offrir en situation d'échec une aide à la décision présentant l'ensemble des solutions retenues par le système, en précisant pour chacune d'elle les conditions déclarant cette

solution envisageable, les influences (avantages et inconvénients) sur l'état courant du simulateur. En qualité d'expert, l'utilisateur peut choisir d'appliquer une des solutions du système, d'adapter une des solution pour son cas particulier ou même de tenter une autre alternative inconnue ou négligée par le système d'aide.

- Offrir pour chaque élément manipulé (paramètre, commande, notion,...) une aide contextuelle précisant la définition, les conditions d'utilisation de l'élément. C'est ce que nous appelons par la suite la condition d'activité selon le contexte. Ainsi chaque élément peut signaler s'il est utile dans le contexte courant. Ainsi plutôt que de guider l'utilisateur pour atteindre un objectif fixé, nous lui proposons d'étudier ce qu'il peut faire dans un état donné. La logique d'utilisation permet de repérer l'état courant du simulateur, de suivre l'évolution du contexte, en observant les actions (via menu) de l'utilisateur.

- Offrir pour chaque élément manipulé au cours de la simulation des informations générales décrivant le rôle de l'élément, les éventuels liens avec d'autres éléments (lien de calcul de valeur, lien de contraintes). Ces explications représentent la couche d'information relative à la logique de fonctionnement.

4- La maquette

La première motivation de la maquette est d'avoir un support pour recueillir les réactions des utilisateurs. Il est indispensable de vérifier en quoi notre approche permet de répondre à l'attente des concepteurs, et ce le plus tôt possible. La présentation de la maquette déclenche des critiques, des demandes des utilisateurs qu'il n'est pas facile d'obtenir par l'échange de discussions. Bien souvent les besoins ne sont pas formalisés par l'utilisateur, et ce dernier ne les formule qu'en réaction aux propositions concrètes.

Le deuxième intérêt est de formaliser les connaissances acquises sur l'utilisation de l'outil de conception, le simulateur SMASH [Smash95]. Ces connaissances sont d'une part acquises par une utilisation personnelle de l'outil (en passant du stade néophyte, au stade avancé), et d'autre part grâce à un échange avec des concepteurs expérimentés et avec l'équipe de développement de l'outil de simulation. Le premier travail est de représenter ces connaissances afin d'ébaucher une structure du système d'aide. Le but est de valider cette représentation des connaissances pour une exploitation par le système d'aide.

Pour des raisons de rapidité d'implémentation, la maquette est totalement dissociée du simulateur. Au cours d'une démonstration, les actions utilisateurs exécutées dans l'outil sont manuellement reportées dans le système d'aide. Ces échanges de données permettent de suivre l'évolution du contexte, et d'offrir une aide contextuelle.

Nous présentons tout d'abord la structure de la maquette, basée sur les principaux états du simulateur perçus par l'utilisateur. Nous détaillons ensuite la représentation des connaissances avant d'illustrer le comportement de l'aide avec un exemple.

4.1- La structure de la maquette

La proposition est de modéliser l'outil par une structure calquée sur la vision offerte à l'utilisateur. A priori, trois types d'analyses peuvent être demandés sur un circuit déjà modélisé : une analyse en régime temporel, une analyse en régime fréquentiel, et une analyse en régime permanent. Cependant, l'utilisateur se voit proposer un ensemble d'analyses plus important, nommément la recherche de l'état de repos du circuit, la mise sous tension du circuit, l'étude temporelle, l'étude fréquentielle et l'étude en régime permanent. Ainsi, des analyses conceptuellement proches mais utilisées pour des besoins particuliers sont-elles distinguées. Par exemple, une analyse de recherche d'état de repos d'un circuit est un cas

particulier d'une analyse en régime permanent, où les signaux d'entrées appliqués correspondent aux alimentations du circuit. Ces deux analyses sont rigoureusement identiques du point de vue algorithmique, mais correspondent à deux étapes de la simulation. La notion d'état de repos, dit aussi point de fonctionnement, est un état de départ pour une analyse temporelle, et un état de référence pour une analyse fréquentielle autour duquel le circuit est linéarisé. Il existe donc des notions d'enchaînements entre les différentes analyses.

Le schéma d'états, [fig1], illustre les différentes étapes et leurs enchaînements au cours d'une session de simulation d'un circuit. S'ajoutent ensuite, entre les différentes analyses, les liens qui illustrent un moyen non standard de mener la simulation. Ces liens correspondent soit à la recherche d'une solution par approximations successives, soit à des analyses qui n'exigent pas le respect des enchaînements explicites. Ce schéma d'état regroupe les informations relatives à chaque analyse en listant exhaustivement l'ensemble des commandes et paramètres actifs, en accord avec le circuit chargé.

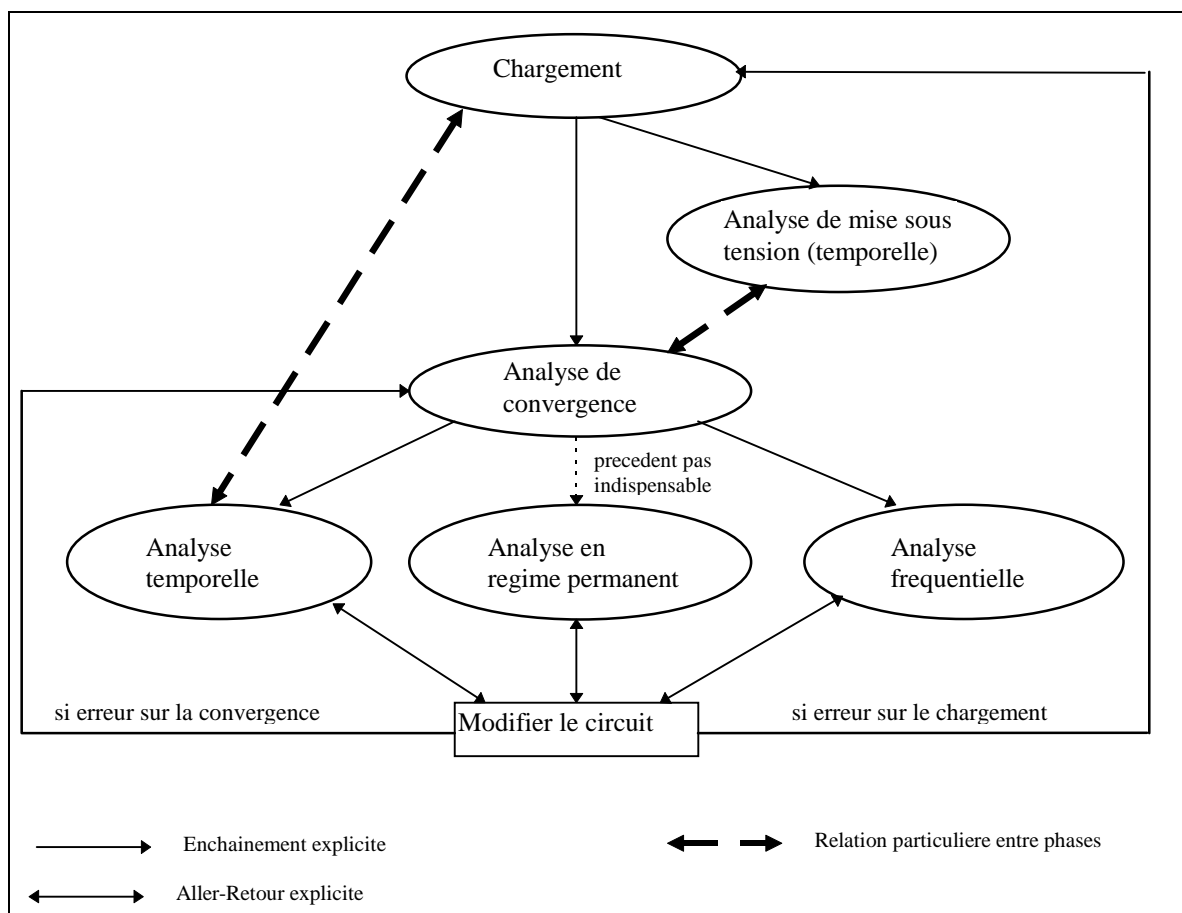


Figure 1 : Schéma d'états de l'application Smash. Le déroulement classique d'une session débute par le chargement du circuit, se poursuit par une analyse de convergence. Selon ses besoins, l'utilisateur peut réaliser une analyse temporelle, une analyse en régime permanent ou une analyse fréquentielle. Après une modification du circuit, la simulation reprend par une analyse de convergence.

4.2- La représentation des connaissances

L'environnement d'implémentation choisi pour la maquette est Shood [Bounaas95], un système de représentation de connaissances qui offre deux niveaux d'instanciation. Ainsi une classe est elle-même instance d'une classe, ce qui permet une description simple des méta-connaissances. La méta-classe permet de déclarer les connaissances structurelles, qui n'évoluent pas au cours du processus modélisé. Au niveau de la classe, les connaissances structurelles sont **instanciées**, et les connaissances contextuelles sont **déclarées**. Ainsi, une instance de classe représente une connaissance contextuelle, avec une nouvelle instance pour tout nouveau contexte. La figure 2 illustre les classes définies pour les phases et pour un ensemble de commandes utilisées pour la maquette.

Le schéma d'états distingue les différentes phases, et identifie les actions possibles et influentes pour chaque type de phase. Il regroupe en fait toute la connaissance structurelle de l'outil. Par exemple, chaque phase est représentée par une classe dont les attributs, déclarés dans la méta-phase, renseignent :

- un descriptif de la tâche associée à la phase,
- les conditions d'accès à la phase,
- la commande d'entrée dans la phase,
- la phase préalable quand un enchaînement est contraint,
- les phases suivantes envisageables,
- la liste des commandes actives,
- la liste des paramètres actifs.

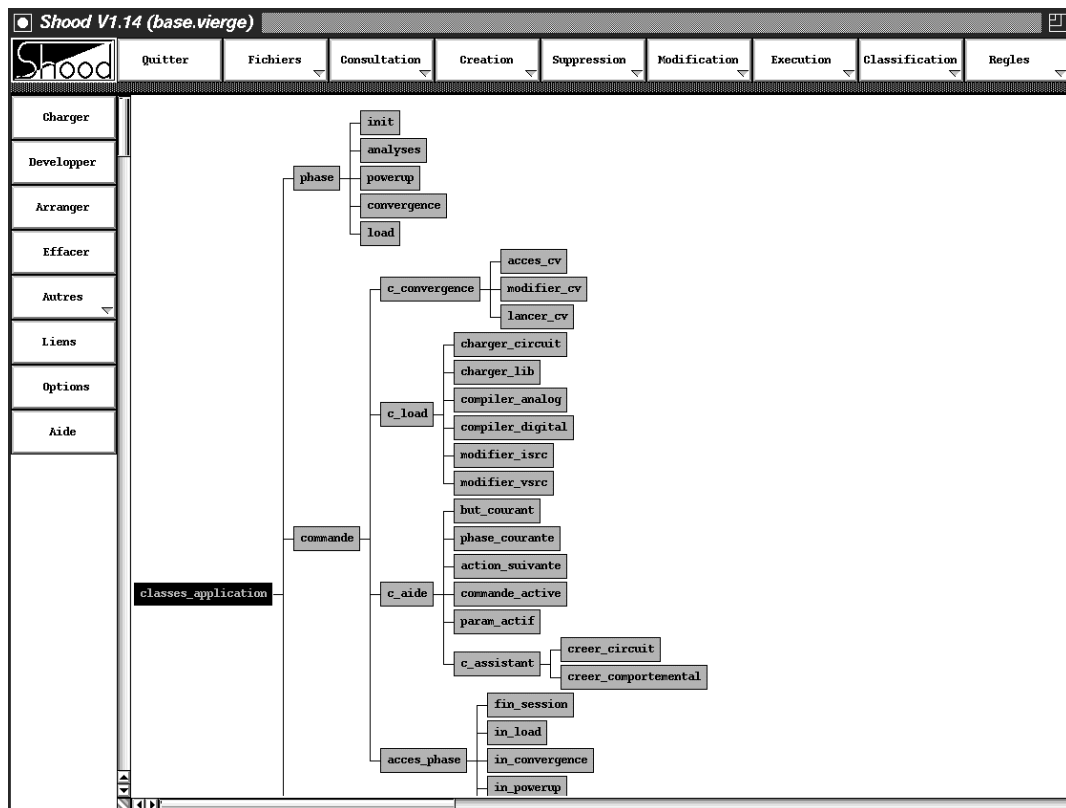


Figure 2 : Sous-graphe des classes manipulées

Au cours d'une session, quand l'utilisateur débute une phase par l'appel de la commande d'entrée, une instance de la classe phase est construite. Les attributs de cette classe ne sont plus généraux, mais renseignent les propriétés courantes. Les commandes et paramètres répondent positivement ou négativement aux règles propres déterminant leur activité dans le contexte. Dans notre application, **le contexte est constitué du circuit, de la phase courante, de la commande courante de l'utilisateur, et de l'historique de navigation dans le schéma d'états.**

4.3- Le comportement de la maquette

Illustrons par un exemple, la notion d'adaptation par règles réflexes vis-à-vis du contexte courant. Les règles réflexes, attachées à chaque élément manipulé au cours de la simulation (paramètre, commande...) reflètent les conditions d'utilisation. Si nous travaillons sur un circuit purement analogique, les paramètres réservés à la partie numérique se déclarent inactifs si l'utilisateur les manipule, car aucun élément du circuit n'est numérique. Au cours de la recherche de l'état de repos du circuit, il est aussi possible que l'utilisateur appelle une commande relative à un composant analogique absent de la liste des éléments du circuit. L'aide signale, en se basant sur les connaissances structurelles de la méta-classe, que la commande est a priori accessible dans cette phase de travail. Cependant, la commande n'est active que si le composant (de type transistor MOS par exemple) est présent. Aujourd'hui les explications sont attachées à chaque paramètre et commande, et sont présentées par un texte laconique. Ainsi la condition pour accéder à l'analyse de mise sous tension [fig1] est placée au niveau de la commande d'accès in-powerup [fig2] et s'énonce : « Cette commande est active si un circuit est correctement chargé ». Tant que les explications restent simples, cette approche est envisageable mais elle devra évoluer. L'historique de navigation permet de savoir quelles phases ont été réalisées avec succès pour ne pas les suggérer à nouveau quand l'utilisateur demande au système d'aide quelle action suivante est envisageable. En effet, l'utilisateur parcourt le schéma d'états de la figure 1 à sa guise et après un chargement, une étude de mise sous tension et une analyse temporelle, il ne faut pas lui suggérer à nouveau l'une de ces phases déroulées avec succès. De manière plus générale l'historique permet de signaler les actions redondantes de l'utilisateur.

4.3.1- L'aide de navigation

L'aide de navigation est chargée d'exploiter le schéma d'états et le contexte pour guider l'utilisateur. Tant que ce dernier ne se heurte pas à un échec, les informations auxquelles il accède donnent les enchaînements possibles, et les actions explicites peuvent être interrogées. L'aide contextuelle et la logique de fonctionnement se complètent pour renseigner l'utilisateur de manière pertinente. La figure 3 est un exemple d'actions proposées à l'utilisateur qui ouvre une session de simulation. Les commandes but-courant, phase-courante, commande-active, param-actif et action-suivante sont les commandes d'aide. Elles expliquent respectivement la tâche courante, le nom de la phase courante, la liste des commandes actives dans le contexte courant, la liste des paramètres actifs et enfin action-suivante permet d'obtenir la suggestion favorisée par l'aide.

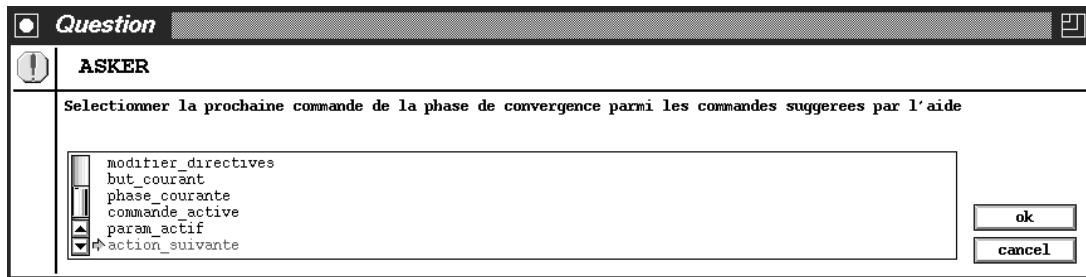


Figure 3 : Actions proposées en début de session

La suggestion favorisée par l'aide est la commande in-load, qui correspond au chargement d'un circuit. Cette suggestion correspond à la première phase envisageable active de la phase d'initialisation. Les autres alternatives sont de créer un nouveau circuit, ou de définir des modèles de composants comportementaux. Ainsi le système d'aide se base sur les méta-connaissances de la phase en cours et sur les règles réflexes pour connaître les éléments actifs dans le contexte courant.

4.3.2- L'aide à la décision

La notion d'aide à la décision est déjà mise en valeur dans l'aide à la navigation, mais notre proposition d'explication est mieux illustrée lors d'un échec de simulation. Lorsque l'utilisateur étudie l'état de repos de son circuit analogique, le simulateur ne parvient pas toujours à obtenir une solution numérique respectant les paramètres de contrôle. Le message d'échec « Unable to converge! » est alors affiché par le simulateur. Ce message est transmis au système d'aide qui se place en situation d'échec. Les figures 4 et 5 illustrent respectivement la réponse à la demande d'aide action-suivante, et à l'appel du descriptif pour la solution cv_diminuer_deltav. La fenêtre [fig4] explique la signification du message et rappelle les paramètres de contrôle : plage de recherche, pas d'itération, et tolérance. Elle sélectionne ensuite deux solutions envisageables. L'utilisateur peut sélectionner l'une des solutions et demander son descriptif. Au niveau de la phase de convergence, qui correspond à l'analyse de convergence de la figure 1, il existe un ensemble de solutions à étudier sur un échec « Unable to converge ». La liste exhaustive présente dans la maquette regroupe :

- cv_élagir-recherche, qui consiste à élargir la plage de recherche en tension au-delà des alimentations du circuit. Ici cette solution n'est pas active car le circuit ne présente pas de tensions hors de la plage fixée.
- cv_diminuer_deltav, qui consiste à diminuer le pas d'itération, et la solution est active.
- cv_par_défaut, qui consiste à mettre tous les paramètres de contrôle à leur valeur par défaut. Comme les paramètres courants ont leur valeur par défaut, la solution n'est pas active.
- cv_relacher_tolerance, qui consiste à relâcher les paramètres de contrôle de tolérance. Comme les courants et tensions du circuits ne sont pas élevés, la solution n'est pas active.
- cv_deltav_0, qui consiste à augmenter lentement le pas d'itération. Cette solution est proposée.
- cv_deltav_-1, qui offre un contrôle particulier du pas d'itération adapté quand le circuit présente des transistors mos. Cette solution n'est pas active.

Le tri parmi l'ensemble des solutions, a priori possibles d'après les méta-connaissances de la phase de convergence, se fait par les règles réflexes déclarant les solutions actives ou non.

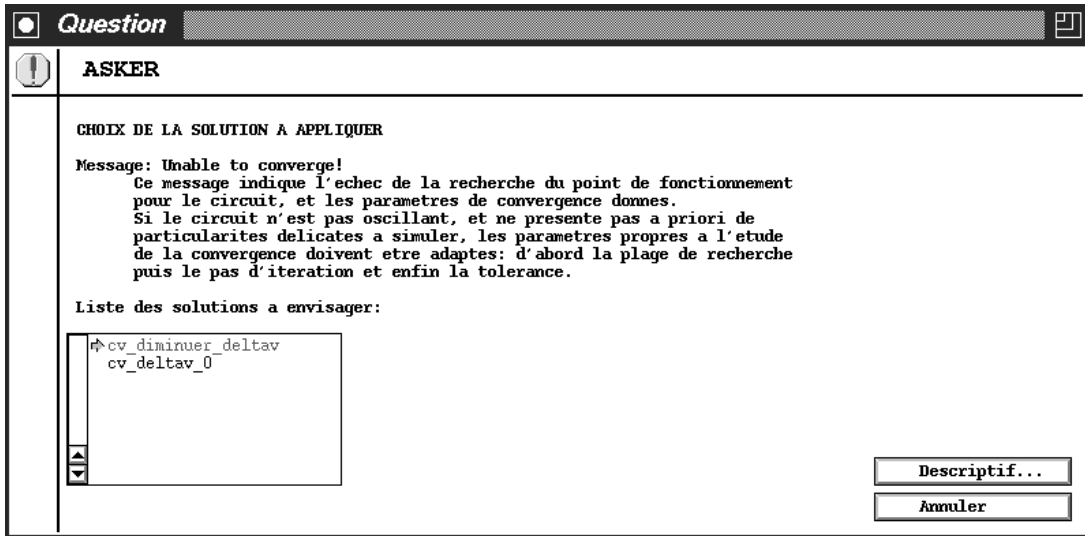


Figure 4 : Suggestions lors d'un échec de convergence

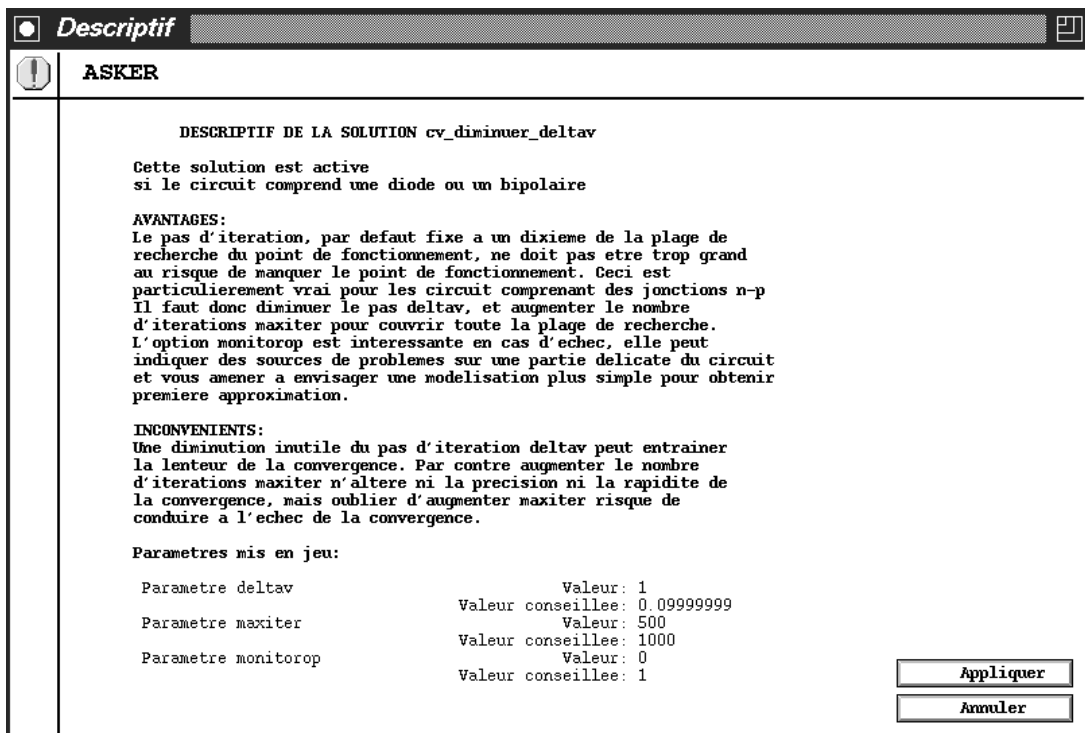


Figure 5 : Avantages et inconvénients d'une suggestion

Le descriptif présenté en figure 5 correspond à la suggestion cv-diminuer-deltav. Il indique tout d'abord quelle condition a permis de rendre cette suggestion active, spécifie ensuite les avantages et inconvénients de cette solution, avant de proposer la solution en terme de modification de paramètres. Ce descriptif est composé par la condition textuelle propre à chaque instance de solution qui explique la règle réflexe, par les influences textuelles qui présentent les avantages et inconvénients. Ces informations sont figées dans la maquette, c'est

--à dire qu'elles ne sont pas construites d'après le contexte. Il n'est pas gênant aujourd'hui d'offrir le même texte qui doit être suffisamment général pour faire comprendre la solution proposée. Par contre, les modifications de paramètres tiennent compte du contexte. Pour chaque paramètre, la valeur indique la valeur courante retenue au niveau du simulateur, et le système d'aide propose une valeur conseillée. Cette valeur est calculée par une règle attachée à la solution (si le pas d'itération diminue, multiplié par deux le nombre d'itération), éventuellement par l'intermédiaire d'un élément ayant rendu cette solution active. Ainsi la présence de transistors bipolaires active la solution `cv_diminuer_deltav`, et propose la valeur de 0.1 pour le paramètre `deltav`.

Si la qualité des explications est satisfaisante pour comprendre les conseils de l'aide, la maquette ne présente toutefois pas tous les aspects souhaités. Ainsi les solutions proposées par le système d'aide ne sont pas flexibles au niveau de la maquette. Pour permettre le suivi des actions de l'utilisateur, la maquette n'autorise pas d'adaptation de sa solution par l'utilisateur. Si la solution « imposée » n'est pas appliquée, mais par exemple est accentuée en contraignant plus les paramètres, le système d'aide ne reconnaît pas la solution. Si l'analyse de convergence lancée avec la suggestion personnalisée échoue à nouveau, le système ne met pas à jour l'ensemble des solutions encore envisageables, et il va proposer à nouveau cette suggestion quand il devrait l'identifier comme inactive.

4.4- Critiques de la maquette

La maquette a été présentée à des utilisateurs occasionnels, en parallèle d'un exemple de simulation délicate. Il faut noter que certains d'entre eux, reconnus experts pour la phase de convergence sur des circuits analogiques, ont découvert des possibilités offertes par le simulateur pour obtenir une solution. Ces solutions sont documentées dans un chapitre qui présente l'ensemble des possibilités en cas d'échec. Il est donc important, même pour les utilisateurs expérimentés, d'offrir des conseils contextuels au cours de la recherche de solution pour fixer ces informations par une mise en pratique. Cette observation illustre l'absence de niveau d'expertise pour dispenser une aide, et nous encourage à laisser l'utilisateur choisir lui-même les explications dont il a besoin.

Le contenu des explications n'a pas été critiqué, par contre la formulation a soulevé des remarques. Les concepteurs, occasionnels ou expérimentés, confrontés à la maquette ont émis le souhait d'un vocabulaire plus physique, partagé par tous les concepteurs. Il est vrai que certains termes, propres à la simulation, ne facilitaient pas la compréhension. Ainsi les explications contiennent le terme « analyse de convergence », qui n'est pas aussi évocateur que l'expression « recherche du point de fonctionnement du circuit », ou « recherche de l'état de repos du circuit mis sous tension ». Pour répondre aux différents utilisateurs, leur proposition est de s'appuyer sur les termes relatifs au circuit physique, aux études et tests faits sur un circuit prototype.

Il nous semble que la nature des explications pourra aussi être enrichie d'informations dédiées aux néophytes, qui mettront en évidence les liens entre les différents paramètres qui sont attachés à une même notion (contrôle du pas d'un algorithme, analyse spécifique...). Ces explications sont en effet souvent demandées par des étudiants découvrant le simulateur.

L'exemple d'aide à la décision offre aux concepteurs occasionnels une solution complète, en indiquant un ordre de grandeur pour les paramètres à modifier. Ces indications de valeurs complètent bien les descriptions textuelles, car si elles offrent une solution peut être imparfaite, elles n'obligent pas l'utilisateur à faire des choix sans support de décision.

Il est intéressant de noter que le manque de flexibilité dans la description de la solution était mal accepté par les concepteurs expérimentés confrontés à la maquette. Bien que la solution proposée par le système soit satisfaisante pour le circuit retenu en démonstration, le souhait d'une personnalisation était exprimé. Cette remarque confirme l'intérêt de notre approche de l'aide pour la conception de toujours offrir un statut d'expert au concepteur.

5- Les perspectives

La représentation des connaissances sur l'outil doit à terme faire coopérer une connaissance formalisée par les experts, souvent sous forme de règles, et une connaissance empirique détenue par des utilisateurs qui n'ont pas le souci de formaliser cette connaissance. Il faut noter que la solution empirique n'est pas facilement obtenue par les règles des experts, et qu'elle ne peut pas être négligée car elle répond au besoin de l'utilisateur et correspond à un savoir faire. La proposition est de permettre, sur un site client, l'apport de cette connaissance empirique et propre au client dans le système d'aide. Bien entendu, cet apprentissage ne doit pas compromettre la cohérence des règles des experts qui constituent la base de l'explication. Comme la connaissance empirique n'est pas a priori expliquée, ou tout au moins n'est pas validée par des experts, il ne faut pas mélanger ces niveaux de connaissances, mais les faire coopérer. Il apparaît alors intéressant de traiter les connaissances empiriques comme des cas, et de les indexer pour les retrouver au moment opportun : cela correspond à la phase de recherche dans le raisonnement à partir de cas où la difficulté réside essentiellement dans la représentation de connaissances (que constitue un cas?) et dans le choix d'index pertinents pour retrouver des solutions adaptées au contexte. L'utilisation du raisonnement à partir de cas (ou Case-Based Reasoning) pour résoudre des problèmes d'outils informatiques n'est pas nouvelle, et il existe aujourd'hui plusieurs systèmes de support (hot-line) basés sur le CBR, [Shimazu94], [Barletta93]. L'intérêt est d'intégrer dans notre système la possibilité de mémoriser des problèmes déjà rencontrés, avec leurs solutions, afin d'ouvrir notre système vers une aide de support complétée de manière interactive [Rewari93].

Le système d'aide proposé sera donc hybride : les règles définissent les connaissances générales, qui permettent d'appréhender le contexte et d'expliquer les suggestions vis-à-vis du contexte, et les cas sont utilisés pour les situations particulières ou plus spécifiques que la connaissance générale. Là encore cette approche n'est pas nouvelle [Golding91], mais elle semble particulièrement bien adaptée au problème de l'aide.

Bibliographie

- Macintosh™, aide du système 7.5.
- [Barletta93], Barletta et Kriegsman, Building a case-based help desk application, IEEE Expert, pp18-26, décembre 1993
- [Berthomé95], A Berthomé-Montoy et J-M Fouet, Utilisation de métaconnaissances pour adapter l'aide à l'utilisateur, INFORSID XIII Congrès, Grenoble, pp329-342, 30 mai au 2 juin 1995
- [Bounaas95], Fethi Bounaas, La représentation de connaissances, une approche par les règles, thèse en informatique, ensimag INPG, octobre 1995
- [Cawsey92], Alison Cawsey, Explanation and interaction, the computer generation of explanatory dialogues, the MIT Press, Cambridge, Massachusetts, London, 1992
- [Chin89], D.Chin, Knome : modeling what the user knows in uc, dans Kobsa & Wahlster editors, User Models in dialog systems, Springer Verlag, Berlin, pp74-107, 1989
- [Forslund95], Goran Forslund, Toward cooperative advice giving system, IEEE Expert Vol 10 n° 4, pp56-62, août 1995

- [Golding91], Golding et Rosenbloom, Improving rule-based system through case-based reasoning, AAAI91, 1991
- [Maïs88], Chantal Maïs, Pour des systèmes d'aide à la réalisation de procédures sous-optimales, ERGO-IA88, pp174-187, 1988
- [Richard83], Richard, Logique de fonctionnement et logique d'utilisation, Rapport de recherche Inria n° 202, avril 1983.
- [Rewari93], Rewari, AI for customer Service and support, IEEE Expert, décembre 1993
- [Senach87], Bernard Senach, Meta-communication, gestion de contexte et modélisation cognitive de l'utilisateur: trois perspectives pour améliorer l'intelligence des systèmes d'aide à l'utilisation, Bulletin de liaison de la recherche en informatique et en automatique, n° 115, pp20-30, 1987
- [Shimazu94], Shimazu, Shibata and Nihei, Case-based retrieval interface adapted to customer-initiated dialogues in help desk operations, AAAI94, pp513-518, 1994
- [Smash95], manuel d'utilisation SmashTM version 3.1, Dolphin Integration, 1995.
- [Tarby94], J-C Tarby, Application de la logique d'utilisation à l'aide contextuelle, ERGO-IA94, Eds l'I.D.L.S, pp181-190, 1994