Interface tactile pour la saisie guidée de connaissances

Jean-Philippe Poli, Jean-Paul Laurent

▶ **To cite this version:**

Jean-Philippe Poli, Jean-Paul Laurent. Interface tactile pour la saisie guidée de connaissances. 27ème conférence francophone sur l'Interaction Homme-Machine (IHM '15), Oct 2015, Toulouse, France. pp.a1, 10.1145/2820619.2820620 . hal-01218446

HAL Id: hal-01218446
https://hal.science/hal-01218446

Submitted on 21 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interface tactile pour la saisie guidée de connaissances

Jean-Philippe Poli, Jean-Paul Laurent
CEA, LIST
91191 Gif-sur-Yvette cedex, France.
jean-philippe.poli@cea.fr, jean-paul.laurent@cea.fr

RÉSUMÉ
Au cours de ces dernières années, les outils d'intelligence artificielle se sont démocratisés et sont de plus en plus souvent utilisés par des personnes qui ne sont pas expertes du domaine.

Parmi ces outils d'intelligence artificielle, les systèmes à base de règles ou de contraintes nécessitent la saisie de l'expertise humaine afin de reproduire le comportement souhaité. Malgré l'explosion des nouveaux périphériques et de nouveaux paradigmes de saisie, comme les tablettes et autres interfaces tactiles, l'ergonomie de ces outils semble ne pas avoir profité de toutes ces avancées récentes.

Dans cet article, nous prenons l'exemple d'un système expert flou pour lequel il faut rédiger des règles. La particularité de la logique floue est que ces règles sont construites d'une manière plus proche du langage naturel qu'en logique classique. Nous présentons notre travail qui consiste en l'exploitation des nouvelles interfaces tactiles afin de rédiger une base de règles floues avec un seul doigt. Nous terminons cet article par l'évaluation de l'interface auprès d'un panel d'utilisateurs.

Mots Clés
Tactile ; système expert flou ; logique floue ; saisie de règles ; base de connaissances ; interaction homme-machine.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces.

INTRODUCTION

Aux prémisses de l'intelligence artificielle (IA), les spécialistes de l'IA et les spécialistes de l'Interaction Homme-Machine (IHM) se sont opposés dans leurs démarches, les premiers souhaitant égaler l'homme et les seconds cherchant à augmenter ses capacités [21]. Aujourd'hui, les outils d'IA se démocratisent et ont pour but d'assister l'homme dans ses tâches quotidiennes et touchent des publics très différents, souvent non spécialistes de ces algorithmes. Alors que les techniques d'IA se sont développées et améliorées au fil des années, ce n'est que récemment que l'on s'intéresse à ce qui a été appelé "l'utilisabilité de l'IA" [12]. En effet, si une technologie doit être utilisée dans un contexte réel par le grand-public, une interaction soignée et efficace avec l'utilisateur est un aspect important de l'acceptation de cette technologie [9]. Par exemple, Dadzie et Petrelli [6] indiquent que malgré des algorithmes de recommandation performants, le site d'e-commerce *Amazon* a dû tester plusieurs méthodes d'affichage des résultats avant de parvenir à une acceptation de l'outil par ses utilisateurs.

Dans cet article, nous présentons notre travail concernant un système à base de règles. Dans de tels systèmes, il est nécessaire de saisir de la connaissance afin qu'un moteur d'inférence puisse reproduire le raisonnement d'un expert humain. La saisie des connaissances est souvent une étape fastidieuse et comme le montre la section Travaux antérieurs, les interfaces de saisie ont peu évolué. Cela explique notre motivation à proposer une interface de saisie de règles exploitant les surfaces tactiles modernes comme les tablettes et un mode d'interaction très répandu, le glisser-déposer (en anglais *drag-and-drop*). En effet, le glisser-déposer sur une surface tactile est une technique d'interaction dont le résultat est directement visible, ce qui permet de solliciter moins d'attention et moins de capacité cognitive [3]. De plus, des études récentes montrent qu'il s'agit d'un geste intuitif qui permet une interaction de grande précision pour des utilisateurs dont les profils et les âges sont variés, notamment grâce au contact continu avec la surface [15]. Ce mode d'interaction nous semble donc intéressant à étudier.

Sans perte de généralité, nous allons illustrer nos travaux par la saisie des règles pour un système expert flou, que nous présentons dans la section suivante. Nous nous intéressons ensuite aux travaux antérieurs ou connexes que nous avons pu trouver dans la littérature. Les sections suivantes sont consacrées à la description de l'outil mis en place ainsi que son évaluation auprès d'un panel d'utilisateurs. Enfin, nous terminerons par une conclusion ouverte sur des perspectives de ces travaux.

CONTEXTE

Un système expert est un composant logiciel basé sur des algorithmes d'intelligence articielle dont le but est de reproduire le raisonnement d'un expert humain dans un domaine donné. L'expertise est rassemblée dans une base de règles. Des algorithmes d'inférence permettent, à partir des entrées présentées au système expert, de produire de nouvelles connaissances (les sorties du système expert). Si

© ACM, 2015. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 27ème conférence francophone sur l'Interaction Homme-Machine, 2015.
http://dx.doi.org/10.1145/2820619.2820620

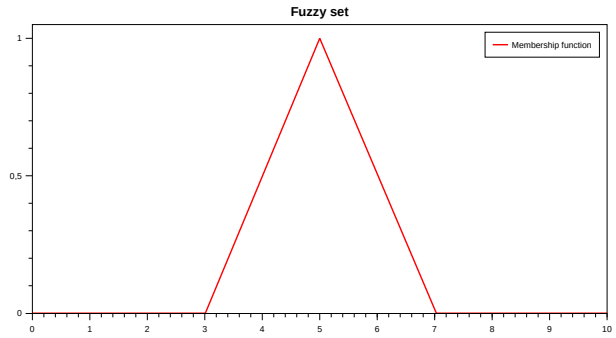

Figure 1. Exemple de sous-ensemble flou défini par une fonction d'appartenance triangulaire. Les abscisses représentent les valeurs du domaine de définition et les ordonnées la valeur d'appartenance.

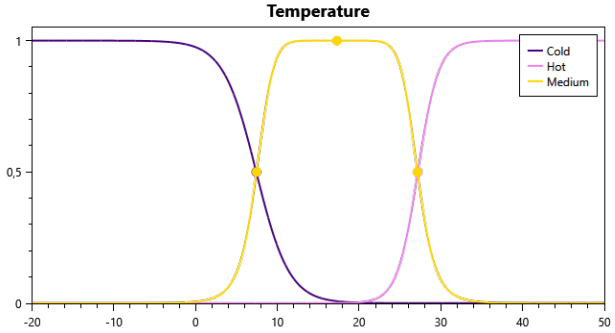

Figure 2. Exemple d'une variable linguistique "température" composée de 3 termes "froide", "moyenne", "chaude" et leurs fonctions d'appartenances respectives. Les abscisses représentent le domaine de définition de la variable et les ordonnées la valeur d'appartenance.

les systèmes experts classiques nécessitent des règles basées sur les logiques classiques, les systèmes experts flous nécessitent des règles dans un formalisme différent sur lesquelles seront appliqués des algorithmes d'inférence adaptés.

La logique floue est une logique multi-valuée introduite par Zadeh en 1965 [22] dans le but de prendre non seulement en compte l'incertitude liée aux entrées du système mais aussi l'imprécision liée au langage. Dans la suite de cette section, nous allons décrire la construction des règles floues, mais nous allons laisser de côté la description des mécanismes d'inférence dont la compréhension n'est pas nécessaire pour la suite. Toutefois, le lecteur curieux peut se référer à [2] pour une introduction plus détaillée.

Pour comprendre comment sont construites les règles en logique floue, il faut dans un premier temps introduire le concept principal sur lequel repose cette logique : les sous-ensembles flous.

Sous-ensembles flous

Définition 1 (Sous-ensemble flou) *Soit X l'univers du discours. Un sous-ensemble flou A sur X est totalement caractérisé par une fonction $\mu(A) : X \to [0, 1]$, appelée fonction d'appartenance (notée dans les figures MF pour membership function).*

La fonction d'appartenance caractérise à quel point un élément x appartient à X, contrairement à la logique classique dans laquelle un élément appartient ou non à un ensemble donné.

La Figure 1 représente un exemple de sous-ensemble flou sur l'intervalle $[0, 10]$ caractérisé par une fonction d'appartenance triangulaire, qui pourrait signifier "à peu près 5".

Le concept de sous-ensemble flou a permis d'introduire plus tard celui de variable linguistique [23].

Définition 2 (Variable linguistique) *Une variable linguistique est un triplet (V, X, T_V) tel que :*

- *V est le nom de la variable (par exemple "température"),*
- *X est le domaine sur lequel la variable linguistique est définie (par exempe, $\mathbb{R}$ ou $[0, 10]$),*
- *$T_V = \{T_1, T_2, \dots\}$ est un ensemble fini de sous-ensembles flous appelés termes qui qualifient V (par exemple, "froide", "chaude").*

Chaque sous-ensemble flou T_i de T_V est associé à un nom et est appelé terme linguistique.

La Figure 2 représente les fonctions d'appartenance des termes "froide", "moyenne" et "chaude" d'une variable linguistique appelée "température" et définie sur $[-20, 50]$.

Règles floues

Les définitions précédentes sont utilisées par les systèmes experts flous pour formaliser le raisonnement humain à travers un ensemble de règles. Ces règles sont formées à partir d'une combinaison de propositions floues élémentaires.

Définition 3 (Proposition floue élémentaire) *Une proposition floue élémentaire est la définition de `V est A` à partir de la variable linguistique (V, X_V, T_V) où A est un des termes de T_V.*

Par exemple, à partir de la variable linguistique "température" illustrée sur la Figure 2, il est possible de définir la proposition floue `température est froide`. Cette proposition est évaluée à partir de sa fonction d'appartenance μ_{cold} : pour une température particulière t, la valeur de vérité de la proposition `température est froide` est donnée par $\mu_{froide}(t)$. La valeur d'une proposition floue appartient à l'intervalle $[0, 1]$ dans $\mathbb{R}$, contrairement à la logique classique dans laquelle les valeurs de vérité sont soit 0 (faux) soit 1 (vrai).

Définition 4 (Expression floue) *Une expression floue est une composition d'un ensemble de propositions floues ou d'autres expressions floues. La composition se fait à l'aide d'un opérateur logique : la négation(non), la conjonction*

(et) et la disjonction (ou), ou encore des opérateurs plus sophistiqués (spatiaux, temporels, ...).

Les propositions floues sont des cas particuliers d'expressions floues. Par exemple, soit deux propositions floues `V est A` à partir (V, X_V, T_V) et `W est B` à partir de (W, X_W, T_W). Ainsi, `V est A et W est B`, `V est A ou W n'est pas B` sont des expressions floues. La valeur de vérité d'une expression floue est obtenue par l'application de tous les operateurs sur les valeurs de leurs opérandes.

La logique floue possède différentes formulations des opérateurs de conjonction (le "et") et de disjonction (le "ou") qui sont des généralisations des opérateurs logiques classiques.

On peut à présent définir les règles floues.

Définition 5 (Règles floues) *Une règle floue est composée d'une prémisse et d'une conclusion et est de la forme*

`SI prémisse ALORS conclusion`.

Une prémisse est une expression floue et la conclusion est une déclaration qui peut prendre différentes formes, la plus courante ressemblant fortement à une proposition floue :

```
SI température est froide ALORS
      chauffage est fort.
```

TRAVAUX ANTÉRIEURS

Dans cette section, nous nous intéressons aux travaux d'IHM qui ont été menés antérieurement et qui sont applicables à la saisie de règles dans un système expert flou. Comme nous avons pu le voir dans la section Contexte, pour définir une règle, il faut à la fois définir l'expression qui sert de prémisse et la conclusion. Cependant, la saisie des variables linguistiques est un problème à part entière, comprenant le problème de définition des fonctions d'appartenance, que nous n'évoquons pas dans cet article. Nous considérons que les entrées et les sorties du système sont connues et définies au préalable afin de nous focaliser sur l'écriture de la règle elle-même.

Les méthodes issues de la littérature peuvent être classées en deux grandes familles selon qu'elles consistent en une représentation textuelle ou graphique, la parole n'ayant, à notre connaissance, jamais été utilisée.

Les méthodes basées sur le texte reposent sur les techniques de traitement du langage naturel [13] et consistent à extraire directement les règles d'un texte. D'un point de vue utilisateur, ces méthodes ont l'avantage de proposer la rédaction des règles directement dans la langue de celui-ci, avec son propre vocabulaire. Le texte est alors analysé et les relations causales sont extraites afin d'obtenir les règles. Ces méthodes ont été appliquées à l'extraction de règles floues concernant le milieu infirmier [17] ou sur des textes en ligne [10]. La compréhension d'un texte libre est encore un sujet de recherche et ces techniques d'extraction de règles sont donc assez limitées : il est donc tout à fait possible que l'utilisateur saisisse un texte dépourvu de sémantique. Pour palier cette difficulté, il est possible de contraindre l'utilisateur en exploitant des techniques d'auto-complétion [7]. Dans un style plus contraint encore, des langages spécifiques peuvent être introduits : on parle alors de langage intermédaire dont la syntaxe est plus ou moins contrainte, comme les travaux de Acampora et Loia sur un langage de représentation des règles floues [1] basé sur XML. Toutefois, la dextérité des utilisateurs sur les claviers virtuels est moindre par rapport aux claviers physiques [11].

Dans un autre esprit, les méthodes graphiques proposent des représentations moins chargées en texte et symbolisant davantage les liens logiques entre les éléments manipulés. C'est le cas des diagrammes de flux (*flowchart* en anglais) avec lesquels on peut représenter chaque partie d'une règle par des boîtes différentes, potentiellement distinguables par leurs formes et leurs couleurs. L'utilisateur est invité à relier chacune de ces boîtes pour marquer le lien logique qui les relie. La Figure 3 montre un exemple de diagramme de flux utilisé pour l'écriture d'une règle : la règle est composée de deux propositions, chacune reliée à une entrée et à un opérateur binaire, etc. La règle affichée est de la forme "`SI` $proposition_1$ `OPERATEUR` $proposition_2$ `ALORS` $conclusion$". Les techniques de saisie de règles basées sur les diagrammes de flux sont à la fois des sujets de recherche comme dans les travaux de Mosconi et Porta [14] et également des outils que l'on trouve dans les systèmes à base de règles (en logique classique) commerciaux tels que *ILOG* d'IBM et *Visual Rule Modeler* de BOSCH. Les diagrammes de flux peuvent facilement être manipulés sur des interfaces tactiles. De plus, si leur lecture est aisée et naturelle, il semble également qu'ils permettent de saisir des règles qui sont plus complexes à exprimer textuellement [4]. Cependant, les utilisateurs ne sont pas tous égaux devant les diagrammes de flux qui semblent nécessiter un niveau d'abstraction que l'on retrouve facilement chez les ingénieurs mais moins chez les autres personnes (voir section Evaluation).

Les matrices d'association sont un autre moyen de saisir des règles dans un système sous la forme d'un tableau à deux dimensions. Ces interfaces prennent deux entrées et proposent une association visuellement simple entre l'occurrence des valeurs de ces deux entrées et la conclusion de la règle. La Figure 4 montre un exemple de saisie de règles floues avec une matrice d'association considérant deux variables linguistiques P et Q en entrée, dont les termes sont respectivement $P_1, \ldots, P_4$ et Q_1, Q_2, et une variable linguistique C en sortie dont les termes sont $C_1, \ldots, C_8$. Sur cette simple matrice, 8 règles sont désignées :

- `SI` P `est` P_1 `ET` Q `est` Q_1 `ALORS` C `est` C_1 ;
- `SI` P `est` P_2 `ET` Q `est` Q_1 `ALORS` C `est` C_2 ;
- etc...

Les matrices d'association permettent ainsi de saisir un groupe de règles très rapidement. Certains logiciels commerciaux comme *fuzzyTECH* les utilisent comme interface de saisie. L'avantage aussi est de forcer à ce que tous

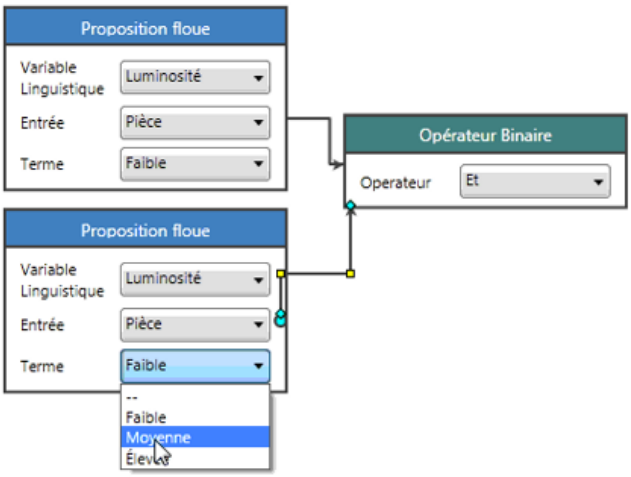

Figure 3. Exemple de diagramme de flux pour la saisie d'une règle floue avec l'outil *Rule flowchart*

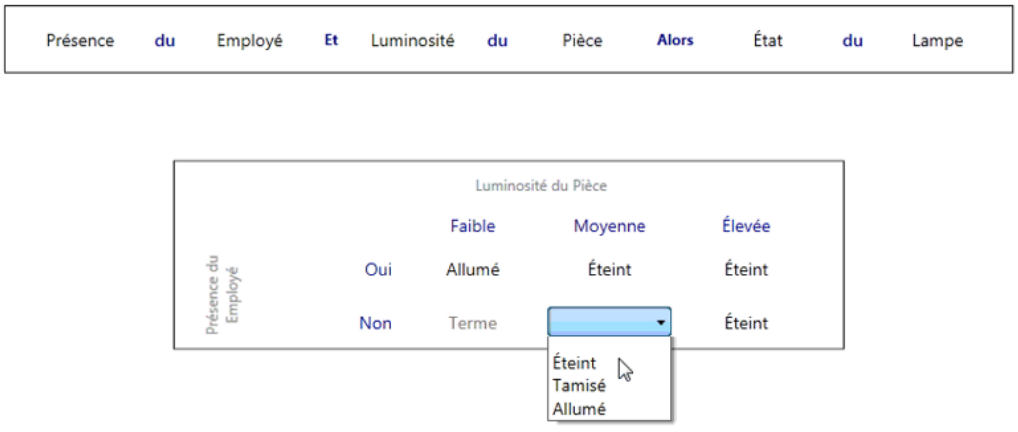

Figure 4. Exemple d'une matrice d'association pour la saisie de 6 règles avec l'outil *Rule Matrix*

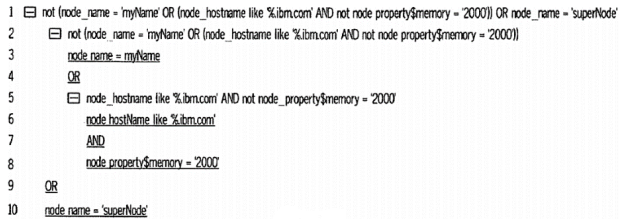

Figure 5. Présentation sous forme d'arbre de la base de règles [18]

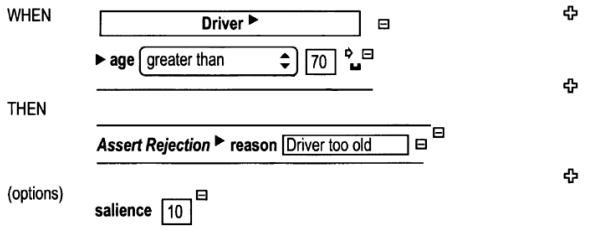

Figure 6. Patron de règles et champs à compléter [16]

les cas soient pris en compte dans la base de règles puisqu'une case vide dans la matrice représente un cas qui ne sera pas traité par le moteur d'inférence. En revanche, les matrices sont limitées de par leur visualisation en 2D qui implique qu'elles ne peuvent traiter que des règles à deux entrées.

Au-delà de ces outils grahiques conventionnels, d'autres interfaces ont été introduites pour améliorer l'expérience utilisateur. On retrouve des outils basés sur le glisser-déposer dans un brevet d'IBM [18] dans lequel la base de règles est représentée sous la forme d'un arbre dans lequel l'utilisateur peut déployer ou cacher les branches (Figure 5) et déplacer les éléments d'une branche à l'autre au moyen d'un glisser-déposer. Malgré cela, les règles ne se lisent plus comme du texte ce qui peut perturber les utilisateurs. Un autre brevet de Red Hat [16] suggère de fournir des patrons de règles dans lesquels l'utilisateur peut choisir les valeurs disponibles (Figure 6). Dans ce cas, la rédaction d'une règle se ramène à choisir un patron de règle et remplir les champs modifiables, ce qui permet une lecture aisée de la règle. Ces patrons sont aussi la limite de l'interface car seulement certaines règles prédéfinies peuvent être rédigées. Le même concept se retrouve dans les travaux de programmation en langage naturel, par exemple appliqués à la domotique. Dans [8, 5], les interfaces permettent de contrôler les actuateurs installés dans un domicile avec une interface intuitive qui permet de construire des règles et des scénarios d'usage. Notons que les termes employés dans les règles sont fortement liés à l'application et que l'on retrouve ainsi les mêmes limitations que les travaux de [16]. Cependant, l'interface étant dédiée à une application donnée, elle représente une manière efficace de saisie des règles.

La section suivante présente nos motivations ainsi que les fonctionnalités de l'éditeur de règles que nous avons mis en place.

EDITEUR DE RÈGLES TACTILE

Motivations

Dans la section précédente, nous avons introduit les méthodes classiques de saisie de règle et des méthodes plus sophisitiquées qui sont transposables au cas particulier de la logique floue que nous mettons en avant dans cet article. Les interfaces de saisie sont généralement un compromis entre l'interprétabilité de la règle par le système et la liberté laissée à l'utilisateur dans la rédaction de la dite règle. De plus, dès qu'elles s'éloignent d'une représentation textuelle, il est possible que l'utilisateur soit décontenancé. Pourtant, les interfaces basées sur le texte sont peu utilisables sur des périphériques tactiles.

Cependant, dans les deux brevets cités ultérieurement, deux idées semblent émerger : d'une part celle d'exploiter le glisser-déposer avec lequel les utilisateurs semblent à l'aise, d'autre part d'avoir des zones à compléter avec des choix prédéfinis contextuellement.

Notre positionnement vis-à-vis de l'état de l'art s'articule autour des nouveaux périphériques avec des écrans tactiles capacitifs. Comme énoncé précédemment, le glisser-déposer sur de tels écrans permettent de viser un public de profils et d'âges différents. C'est également pour cela que nous avons souhaité garder une forme proche du langage naturel tout en proposant des fonctionnalités similaires aux interfaces décrites précédemment.

La difficulté que nous devons prendre en compte est que l'interface de saisie de règles doit être générique et non pas limitée à une application donnée. C'est en effet la philosophie même des systèmes experts qui nous l'impose puisque leur avantage est que le moteur d'inférence est développé une seule fois et qu'il est appliqué à différents domaines, différentes problématiques par le biais de sa base de règles.

Description et fonctionnalités

Dans cet article, nous considérons la rédaction des règles en supposant que les entrées et les sorties du système ont déjà été créées. Pour plus de flexibilité, nous avons séparé le concept de variable linguistique de la logique floue en une entrée et une grandeur mesurée. Cela permettra à terme de factoriser les définitions et d'avoir plusieurs sources pour une même grandeur mesurée.

Notre interface propose de rédiger une règle selon un patron à trous afin que tous les utilisateurs puissent facilement et rapidement produire des règles. Nous appelons ce patron un fantôme, par rapport à l'apparence pâle qu'il prend tant qu'il n'est pas complété. Il est possible de choisir une forme générale de règle, puis les types de propositions qui la composent jusqu'à constituer une règle. Il est également possible de sélectionner et déplacer les différentes parties d'une règle, ou d'insérer un nouvel opérateur. L'insertion d'un nouvel opérateur peut entraîner automatiquement l'insertion d'un fantôme qui décrira la forme que devrait prendre la partie manquante de l'expression courante. Toutes ces opérations se font par glisser-déposer. La Figure 7 présente une vue de l'interface graphique avec une base de règles en cours d'édition, dans

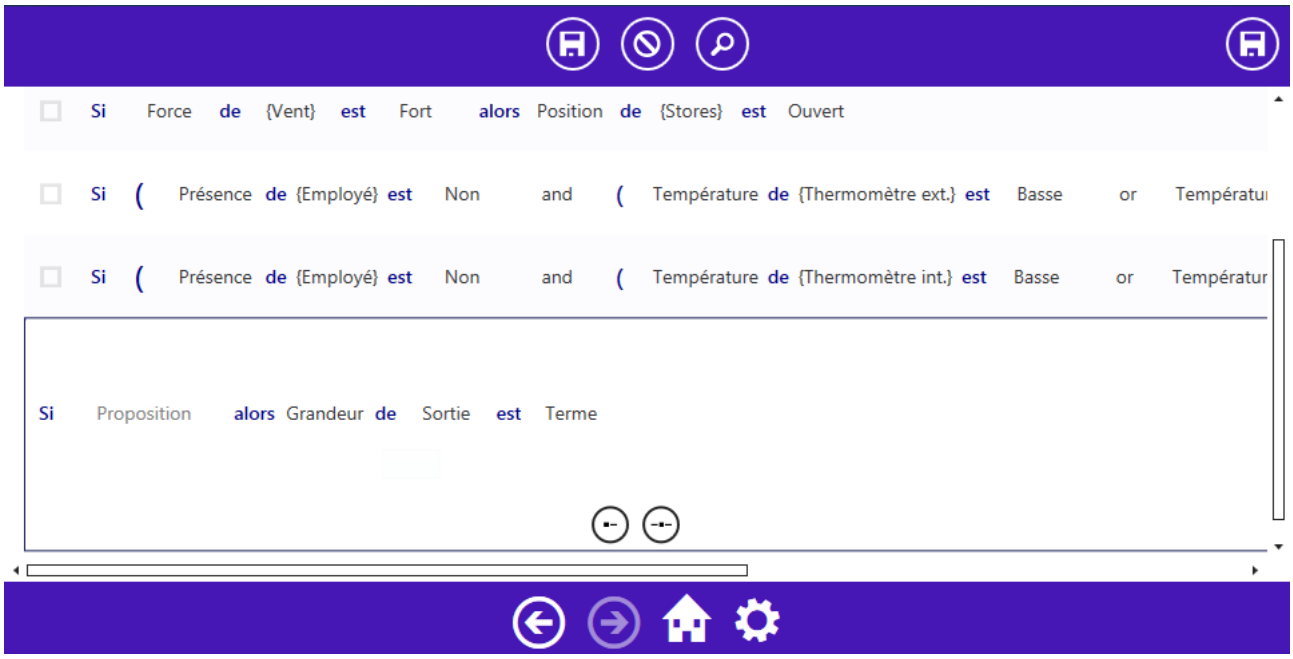

Figure 7. Vue générale d'une base de règles en édition

laquelle apparaît le fantôme d'une proposition, en attendant qu'une forme de proposition soit choisie.

A ce stade, l'éditeur serait trop permissif en permettant de déplacer des opérateurs ou des expressions n'importe où. Nous avons adopté différentes mesures pour que l'utilisateur soit guidé dans les actions qu'il peut faire, sans toutefois trop le limiter dans ses interactions : des *feedforward* permettent de montrer les actions possibles et des *feedback* mettent en valeur le résultat de la dernière action.

Ajout d'une règle

Pour créer une règle, il suffit de cliquer sur le bouton "ajouter" matérialisé par un symbole "+". La règle en cours d'édition est mise en avant par un encadré tout en laissant les autres règles visibles. Elle est matérialisée par un patron contenant un fantôme de proposition pour la prémisse. En cliquant sur le fantôme, un menu circulaire apparaît afin de sélectionner le type de proposition souhaité : soit de la forme `grandeur de entrée est adjectif` soit de la forme `entrée vaut valeur`. Chaque terme personnalisable peut être modifié sur appui, en proposant une liste de choix ou la possibilité d'ajouter du vocabulaire ou des entrées à la base de règles courante. L'éditeur vérifie que la grandeur sélectionnée peut être mesurée à partir de l'entrée choisie et affiche ainsi uniquement les choix possibles lorsqu'une des deux entités est renseignée. La Figure 7 montre un patron de règle avec ses différents fantômes : "Proposition", "Grandeur", "Sortie" et "Terme".

Composition des expressions

Il est possible de créer une expression (unaire ou binaire) en faisant glisser l'icône correspondante vers une portion de la prémisse de la règle. Pour cela, un fantôme d'opérateur est déposé, ainsi qu'un fantôme complétant l'expression ainsi formée si nécessaire. Des flèches rouges animées indiquent où l'opérateur peut être déposé pendant le glisser (Figure 8).

En cliquant sur le fantôme "Op." de l'opérateur, un menu circulaire présente les différents opérateurs disponibles (Figure 9). A la sélection, le fantôme est remplacé par l'opérateur sélectionné. Pour modifier l'opérateur, il suffit de retaper dessus pour faire apparaître le menu circulaire. Les opérateurs sont représentés par des icônes. Si une souris est détectée, le survol de l'icône affiche une aide contextuelle, sinon il suffit d'appuyer de manière prolongée dessus.

Déplacement d'une partie d'une règle

Nous avons opté pour un système de sélection classique basée sur une marquise. Pour cela, il faut vérifier que la partie sélectionnée est cohérente avant d'autoriser son déplacement. Ensuite, des boutons contextuels apparaissent pour copier la portion d'expression ou la supprimer. Il est également possible de faire un glisser-déposer de la sélection à un autre endroit de la règle. Les emplacements possibles sont alors indiqués par des flèches rouges animées comme précédemment (Figure 10).

Dans la section suivante, nous allons évoquer l'évaluation de cet éditeur de règles.

EVALUATION

Nous allons décrire dans un premier temps le protocole expérimental que nous avons mis en place ainsi que le panel d'utilisateurs. Nous terminerons cette section par le dépouillement des résultats.

Protocole

Dans la littérature, certaines méthodes de saisie de règles ont déjà été comparées : nous pouvons citer par exemple la comparaison du diagramme de flux et de la matrice d'association [19] ou encore celle d'un langage structuré basé sur la langue anglaise aux deux précédentes [20]. Dans

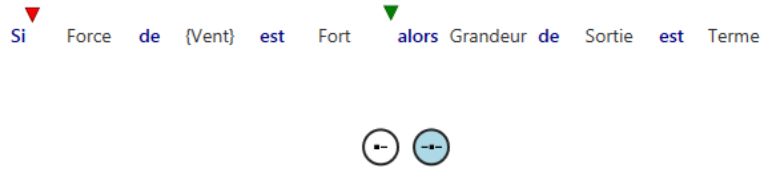

Figure 8. Indication des destinations possibles pendant le glisser-déposer de l'opérateur binaire

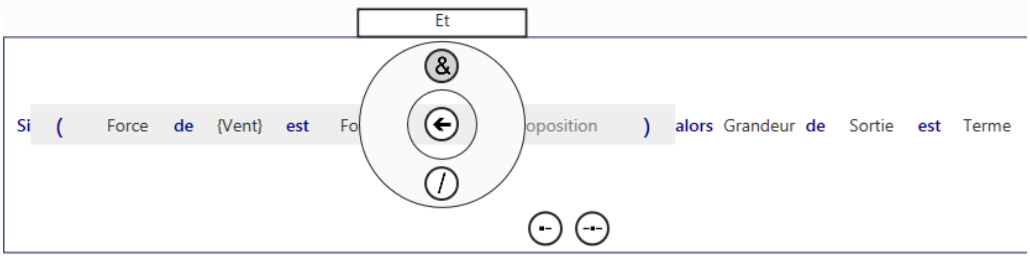

Figure 9. Choix de l'opérateur binaire par menu circulaire

notre cas, nous souhaitons naturellement confronter notre outil à d'autres outils existants. Pour des raisons de disponibilité, notre choix s'est porté sur les diagrammes de flux et les matrices d'association. A partir de librairies existantes que nous avons personnalisées de façon à ce que les apparences soient relativement proches, nous avons créé deux outils (*Rule matrix* et *Rule flowchart*) en plus du notre (*Rule editor*). Cette comparaison s'est faite selon les critères suivants : le temps de prise en main, le temps moyen d'écriture des règles et le nombre de règles erronées. A ces mesures quantitatives s'ajoutent des informations qualitatives issues d'un questionnaire confié aux participants à l'issue du test.

Puisqu'un système expert est un outil applicable à des domaines très variés, nous cherchons à toucher une très grande variété d'utilisateurs. En effet, dans le système expert que nous avons développé, les règles peuvent être écrites par des experts de la logique floue, par des scientifiques qui ne sont pas spécialistes des systèmes experts et de la logique floue mais qui veulent l'intégrer dans un logiciel ou un système plus complexe. Il en va de même pour nos partenaires industriels qui souhaitent formaliser eux-mêmes leurs connaissances ou directement leurs utilisateurs finaux qui veulent configurer leur système.

C'est pour toutes ces raisons que nous avons constitué un panel de 27 utilisateurs assez varié (52% d'hommes et 48% de femmes). Parmi ces utilisateurs, nous avons sélectionné 6 utilisateurs experts (qui utilisent des systèmes experts flous dans leur métier), 8 scientifiques (qui ne connaissent pas les systèmes experts flous) et 13 utilisateurs quelconques, sans aucune culture scientifique.

Nous avons choisi une thématique pour les règles que nous souhaitons qu'ils rédigent : la domotique. Le test consiste à contrôler des volets automatiques en fonction de différents critères observés : la présence de la personne dans la pièce, la luminosité, la force du vent et la température intérieure et extérieure. Le test a été réparti en 3 exercices qui seront résolus par chacune des 3 interfaces :

- Le premier exercice consiste à rédiger des règles basées sur deux entrées seulement. Aucune aide n'est donnée à l'utilisateur quant à l'utilisation des interfaces. Le but est de prendre en main les différents outils et le temps d'accomplissement de l'exercice reflètera en partie le temps de prise en main de l'interface.

- Entre le premier et le second exercice, l'utilisateur qui n'a pas su prendre en main au moins une interface reçoit des explications. L'exercice 2 consiste également à rédiger des règles basées sur deux entrées. Cette fois-ci, le temps d'exécution reflète la rapidité avec laquelle il est possible d'écrire une règle.

- Enfin, le troisième exercice consiste à rédiger des règles plus complexes, que les matrices d'association ne permettent pas de rédiger. L'utilisateur est invité à se rendre compte de cette limitation tout seul.

En pratique, les tests ont duré entre une heure et deux heures selon le candidat, ce qui explique que le panel reste restreint.

Afin de ne pas introduire de biais dans l'évaluation, deux mesures ont été prises. Dans un premier temps, les règles à rédiger ont été données aux utilisateurs en jouant sur les constructions causales de la langue française : par exemple, en employant une construction de la forme "quand le vent est fort, les volets sont remontés" au lieu de "si la force du vent est forte alors la position des volets est remontée". Cela permet de ne pas prendre en compte la difficulté de l'utilisateur à mener un raisonnement logique. Dans un second temps, il a été décidé de présenter les interfaces dans un ordre aléatoire afin qu'une technique ne soit pas discriminée par rapport aux autres.

Les participants ont dû remplir un questionnaire assez détaillé afin de mettre en évidence des facteurs pouvant expliquer leurs performances, mais les tests statistiques ont montré que seul le niveau d'éducation a un réel impact. Par exemple nous demandions aux personnes si elles possédaient un *smartphone* avec écran tactile (dans ce cas,

Figure 10. Options après sélection d'une partie des expressions

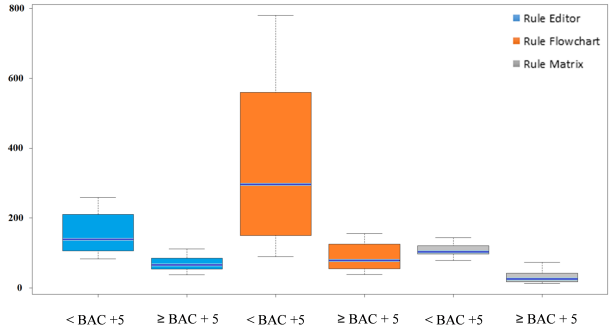

Figure 11. Temps d'exécution de l'exercice 1 (en s) en fonction du niveau d'étude

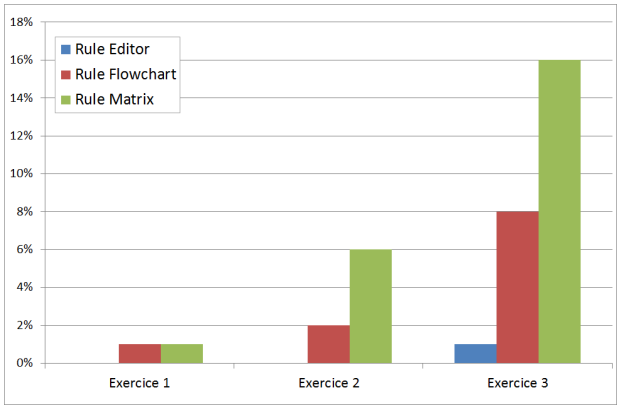

Figure 12. Taux d'erreur de tous les utilisateurs par exercice et par outil (en %)

le glisser-déposer sur surface tactile serait maîtrisé), mais 100% du panel possède un tel terminal.

Analyse des résultats

Nous avons effectué des tests statistiques sur les résultats obtenus aux différents exercices, à la fois en termes de temps et de nombre d'erreurs. Cependant, compte tenu de la population et de l'amplitude des résultats, les tests de Wilcoxon n'ont pas donné de résultats significatifs. Nous nous sommes ensuite intéressés l'application de l'ANOVA (analyse de variance). Pour cela, nous avons procédé à des tests d'homoscédasticité des variances (test de Bartlett) qui indiquent que les variances entre nos différentes classes sont significativement homogènes. Enfin, les résultats de l'ANOVA indiquent une différence significative entre les 3 méthodes de saisie de règles avec un niveau de confiance de 10%.

Plus simplement, la Figure 11 indique les durées de résolution de l'exercice 1 réparties en fonction du niveau d'étude. La figure représente différents éléments statistiques : la moyenne est représentée par un trait épais contenu dans la boîte à moustache. La boîte s'étend du quartile 0.25 au quartile 0.75 et les moustaches s'étendent jusqu'à la valeur distante d'au moins 1,5 fois la distance interquartile. Les durées de l'exercice 1 sont plus longues que pour les autres exercices puisqu'elles comprennent la prise en main de l'outil par le participant qui, nous le rappelons, ne reçoit aucune aide de l'extérieur à ce stade de l'expérimentation. Ces graphiques montrent d'une part la disparité entre les participants au niveau d'étude supérieure inférieur à BAC+5 et les autres : le temps moyen de résolution de l'exercice est toujours plus long. De plus, il y a une plus grande variété dans la rapidité de prise en main des outils pour les personnes dont le niveau d'étude est inférieur à BAC+5. D'autre part, les graphiques nous montrent que le diagramme de flux est l'outil le plus difficile à prendre en main, particulièrement par les gens qui ont un niveau d'étude inférieur à BAC+5, alors qu'au contraire, la matrice d'association permet de résoudre l'exercice très rapidement par tout le monde. Il faut pour bien comprendre ces chiffres rappeler qu'il y a peu de moyens d'interagir avec la matrice : il suffit de choisir deux entrées et de rentrer la valeur dans chacune des cases. Il est donc normal que l'outil soit pris en main rapidement. Enfin, les durées d'exécution sur notre outil sont plus longues que pour la matrice d'association, bien que les durées moyennes soient comparables. Les participants ayant fait de moins longues études ont également eu plus de mal à prendre en main *Rule editor* que *Rule matrix*.

Les durées de réalisation des exercices suivants sont assez similaires si ce n'est que les durées maximum sont inférieures à celles de l'exercice 1. Les matrices d'association restent l'outil le plus rapide pour créer des règles, suivies de près par notre éditeur de règles. Les diagrammes de flux restent encore les plus lents, particulièrement pour les personnes ayant un niveau d'étude inférieur à BAC+5 (en moyenne plus de 120 secondes d'écart entre les deux populations). En termes de catégorie de population, les experts en intelligence artificielle ont réalisé le plus rapidement tous les exercices quelque soit l'interface, suivis par les scientifiques puis par les autres candidats. Ce résultat est une confirmation de l'intuition que l'on peut avoir.

En termes de rapidité de prise en main et d'exécution, rien ne semble plus rapide que la matrice d'association. Cependant, nous avons pour l'instant considéré uniquement le temps d'exécution des exercices, sans regarder les erreurs produites par les utilisateurs. Les tests statistiques montrent que le nombre d'erreurs produites est indépendant de l'éducation des participants.

La Figure 12 montre le taux d'erreur obtenu par tout le panel sur chacun des exercices avec chacun des outils. Les matrices d'association, qui sont le moyen le plus rapide pour saisir des règles, obtiennent le taux d'erreur le plus grand, croissant avec la difficulté de l'exercice. Le fait de naviguer par paire d'entrées pour créer les règles semble rendre difficile l'exercice et nécessite plus de concentra-

tion. De plus, nous comptons comme erreur les règles qui ont été créées par l'utilisateur en essayant d'écrire des règles à plus de deux entrées (ce qui, nous le rappelons, est impossible). Enfin, les règles sont plus difficiles à relire sous forme de tableau que sous une forme plus textuelle. C'est également le cas pour le diagramme de flux, ce qui pourrait aussi expliquer le taux d'erreur de l'outil *Rule flowchart*. En revanche, notre outil *Rule editor* semble avoir permis aux participants de faire moins de fautes : ceci peut s'expliquer par le fait que les règles sont présentées sous une forme facilement lisible par tous, contrairement aux autres outils. Dans le cas de *Rule editor*, une seule faute a été commise.

	Experts	Intermédiaires	Novices
Rule editor	60%	60%	50%
Rule matrix	20%	10%	0%
Rule flowchart	20%	10%	38%
Rule editor + *Rule flowchart*	0%	20%	12%

Table 1. Préférences des utilisateurs pour un outil selon leur connaissance des systèmes experts

D'une manière plus subjective, nous avons demandé aux participants quel était l'outil qu'ils préféreraient s'ils devaient l'utiliser au quotidien chez eux. Tous les participants dont le niveau d'étude est inférieur à BAC+5 ont répondu qu'ils préféraient *Rule editor*. Cela paraît logique puisque notre outil ne repose pas sur une représentation mathématique ou logique auxquelles ces utilisateurs ne sont pas habitués. En revanche, pour les participants ayant fait de longues études, les réponses sont un peu plus variées : 43% ont répondu *Rule editor*, 29% *Rule flowchart*, 9% *Rule matrix* et enfin 17% souhaiteraient une utilisation de *Rule editor* pour des petites et moyennes règles et *Rule flowchart* pour construire des règles plus complexes.

Nous avons aussi projeté les réponses en fonction du niveau de connaissance des systèmes experts flous (experts, intermédiaires, novices) : 60% des experts préfèrent *Rule editor*, les autres se répartissant équitablement entre les deux autres outils. Aucun des non initiés aux systèmes experts n'a apprécié *Rule matrix*, malgré ce que l'analyse quantitative a révélé ; en revanche 50% d'entre eux ont préféré *Rule editor* et 12% un mélange entre *Rule editor* et *Rule flowchart*. Enfin, 60% des connaisseurs intermédiaires ont préféré *Rule editor*. Tous les chiffres sont reportés dans le tableau 1.

DISCUSSION

D'après le dépouillement de l'expérimentation, les matrices d'association restent un outil rapide et facile à prendre en main pour des personnes avec un bon niveau d'études. Cependant, pour des personnes moins familières aux mathématiques et à la logique, la prise en main est moins facile et les fautes sont très nombreuses. De plus, l'outil est très limité car seules des règles à deux entrées et sans négation peuvent être rédigées.

De manière similaire, les diagrammes de flux semblent appréciés par les personnes plutôt scientifiques, surtout pour des règles complexes. Cependant, la relecture n'est pas aisée et les utilisateurs novices ont du mal à raisonner correctement et vérifier leurs règles.

L'éditeur de règles que nous proposons dans cet article semble convenir à la majorité des utilisateurs. La prise en main et la vitesse de rédaction des règles sont similaires à ceux des matrices de confusion. Cependant la forme textuelle semble permettre d'atteindre un taux d'erreur assez bas, voire nul. Toutefois, le temps de rédaction d'une règle n'égale pas celui des matrices si la règle peut être prise en charge par l'interface. Le retour utilisateurs nous a permis de nous rendre compte que les icônes représentant les opérateurs (voir Figure 7) n'étaient pas explicites pour la majorité d'entre eux et qu'il fallait apporter une aide contextuelle directement sur l'interface graphique pour avoir une prise en main encore plus rapide.

Le principe des interactions proposé dans cet article est appliqué, sans perte de généralité, aux systèmes experts flous : en effet, il s'applique également aux autres systèmes experts, notamment aux gestionnaires de règles métiers, souvent utilisés dans les grandes entreprises. D'autres systèmes très présents dans l'industrie utilisent également des connaissances, notamment les solveurs à base de contraintes utilisés par exemple pour la planification de ressources, ou encore les moteurs d'optimisation dont les contraintes sont exprimées par des formules arithmétiques et logiques. Pour cela, il suffit de revoir la grammaire sous-jacente à l'outil.

CONCLUSION

Dans cet article, nous avons proposé un nouvel outil pour la saisie de connaissances, que nous avons illustré par la saisie de règles dans un système expert flou. Les techniques d'interaction modernes sont exploitées pour fournir une expérience utilisateur agréable, une prise en main rapide et intuitive : les tablettes tactiles sont idéales pour exploiter le glisser-déposer qui est une interaction très appréciée des utilisateurs, à laquelle la majorité des personnes est familière.

Nous avons mené une campagne de tests sur 27 utilisateurs aux profils différents afin de comparer notre outil à deux outils majeurs dans le domaine des systèmes experts. Les tests montrent qu'une majorité des utilisateurs apprécient la prise en main et l'utilisation de notre interface graphique et qu'elle permet de par sa manière de présenter les règles de réduire le taux d'erreur dans la saisie des règles.

Confronter l'interface graphique à des utilisateurs nous a permis d'introduire de nouvelles fonctionnalités et d'ajouter de l'aide contextuelle lorsqu'une opération de glisser-déposer commence ou lorsqu'un clic ou un contact est détecté. Depuis ces travaux, l'outil a été utilisé en production par des membres de l'équipe et a permis de saisir une base de 299 règles.

REMERCIEMENTS

Cette recherche a été soutenue par BPI-France dans le cadre du projet EDENS (Eco-District Energy Network Systems).

BIBLIOGRAPHIE

1. Acampora G. & Loia V. Fuzzy markup language: A new solution for transparent intelligent agents. In *Intelligent Agent (IA), 2011 IEEE Symposium on* (2011), 1–6.
2. Bouchon-Meunier B. *La logique floue*. PUF, 2007.

3. Caprani N., O Connor N. E. & Gurrin C. Touch screens for the older use. In *Assistive Technologies*. Fernando A. Auat Cheein, 2012.

4. Chein M., Mugnier M. & Croitoru M. Visual reasoning with graph-based mechanisms: the good, the better and the best. *The knowledge engineering review*, 28 (2013), 28–51.

5. Coutaz J., Demeure A., Caffiau S. & Crowley J. L. Early lessons from the development of spok, an end-user development environment for smart homes. In Proc. *UbiComp '14 Adjunct*, ACM (2014), 895–902.

6. Dadzie A.-S., Lanfranchi V. & Petrelli D. Towards ai usability: Problems, strategies and practicals. In *CHI'08 extended abstracts on Human Factors in Computing Systems* (2008).

7. Darragh J., Witten I. H. & James M. The reactive keyboard: a predictive typing aid. *Computer 23*, 11 (1990), 41–49.

8. Fontaine E., Demeure A., Coutaz J. & Mandran N. Retour d'expérience sur kiss, un outil de développement d'habitat intelligent par l'utilisateur final. In Proc. *Ergo'IHM '12*, ACM (2012), 153:153–153:160.

9. Gajos K. Z. & Weld D. S. Usable ai: Experience and reflections. In *CHI'08 Workshop on Usable Artificial Intelligence* (2008).

10. Hassanpour S., O'Connor M. & Das A. A framework for the automatic extraction of rules from online text. In *Rule-Based Reasoning, Programming, and Applications*, N. Bassiliades, G. Governatori, and A. Paschke, Eds., vol. 6826 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, 266–280.

11. Henze N., Rukzio E. & Boll S. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In Proc. *CHI '12*, ACM (2012), 2659–2668.

12. Jameson, Anthony, Spaulding, Aaron, Yorke-Smith & Neil. Introduction to the special issue on Usable AI. *AI Magazine 30*, 4 (2009).

13. Jones K. Natural language processing: A historical review. In *Current Issues in Computational Linguistics: In Honour of Don Walker*, A. Zampolli, N. Calzolari, and M. Palmer, Eds., vol. 9 of *Linguistica Computazionale*. Springer Netherlands, 1994, 3–16.

14. Mosconi M. & Porta M. A data-flow visual approach to symbolic computing: implementing a production-rule-based programming system through a general-purpose data-flow vl. In *Visual Languages, 2000. Proceedings. 2000 IEEE International Symposium on* (2000), 83–84.

15. Motti L., Vigouroux L. & Gorce P. Drag-and-drop for older adults using touchscreen devices: effects of screen sizes and interaction techniques on accuracy. *Proceedings of the 26th Conference francophone on Interaction Homme-Machine (IHM'14)* (2014), 139–146.

16. Neale M. Natural language enhanced user interface in a business rule management, 2013.

17. Nii M., Yamaguchi T., Takahashi Y., Uchinuno A. & Sakashita R. Fuzzy rule extraction from nursing-care texts. In *Multiple-Valued Logic, 2009. ISMVL '09. 39th International Symposium on* (2009), 30–35.

18. Smith M., Black-Ziegelbein E. & Hee T. Drag and drop rule topology, 2012.

19. Subramanian G. H., Nosek J., Raghunathan S. P. & Kanitkar S. S. A comparison of the decision table and tree. *Commun. ACM 35*, 1 (1992), 89–94.

20. Vessey I. & Weber R. Structured tools and conditional logic: An empirical investigation. *Commun. ACM 29*, 1 (1986), 48–57.

21. Winograd T. Shifting viewpoints: Artificial intelligence and human-computer interaction. *Artificial Intelligence 170* (2006), 1256–1258.

22. Zadeh L. Fuzzy sets. *Information and Control 8*, 3 (1965), 338 – 353.

23. Zadeh L. A. The concept of a linguistic variable and its application to approximate reasoning. Part 3. *Information Sciences 9* (1975), 43–80.