Explicabilité : vers des dispositifs numériques interagissant en intelligence avec l'utilisateur

Pierre-Antoine Champin, Béatrice Fuchs, Nathalie Guin, Alain Mille

Explicabilité : vers des dispositifs numériques interagissant en intelligence avec l'utilisateur

Pierre-Antoine Champin*, Béatrice Fuchs*
Nathalie Guin*, Alain Mille *,**

*Université Lyon, Université Lyon1, CNRS, LIRIS, F-69622
**Coexiscience, F-69100

Résumé. Cet article de positionnement postule que pour qu'un utilisateur et un dispositif technique numérique soient *en intelligence*, il est nécessaire de penser l'explicabilité. Nous proposons une définition de l'explicabilité fondée sur l'objectif d'appropriation de tout dispositif technique numérique par l'utilisateur. L'explicabilité nécessite l'explicite, mais aussi les processus d'exploration des régulations conditionnant le comportement des dispositifs techniques numériques. Après avoir passé en revue ce qui est explicité ou non par les dispositifs concernés (en particulier lorsqu'il s'agit de dispositifs dits intelligents), nous proposons de soutenir les processus d'explication par l'usage des traces d'interaction comme matériau à modéliser avec l'utilisateur jusqu'à pouvoir *aligner* ou *mettre en congruence* sa compréhension avec la réalité du fonctionnement du dispositif, dans le contexte de son activité. Nous soutenons que cette approche est très prometteuse pour améliorer l'explicabilité, y compris lorsque l'explicité n'est pas accessible ou peut-être même n'existe pas à l'avance.

1 Introduction

L'équipe TWEAK a comme objectif de recherche l'étude des conditions permettant aux dispositifs techniques numériques d'interagir *en intelligence* avec l'utilisateur. Au delà du développement de dispositifs *intelligents* il s'agit aussi que ces dispositifs facilitent leur appropriation par les utilisateurs. Cette appropriation suppose un apprentissage encapacitant de l'utilisateur pour qu'il puisse *agir* et adapter le dispositif à son propre contexte, à ses propres connaissances, à ses propres objectifs. La nécessité de cette appropriation poursuit des objectifs d'efficacité dans les usages (Liquète et al., 2012) mais aussi de capacité éthique (Mille, 2019). Pour qu'un processus d'appropriation puisse advenir pendant l'interaction avec un dispositif technique numérique, il est nécessaire que les régulations encapsulées dans ce dispositif, régissant la manière dont il a été préparé pour réagir aux différentes situations rencontrées, soient *explicables* lors de leur mise en œuvre par l'utilisateur.

Nous proposons de considérer les traces d'interaction comme le matériau de départ pour les processus d'explication. En effet, une trace d'interaction contient des éléments combinés issus des fonctions du dispositif et des actions de l'utilisateur. Formalisées, ces traces permettent de mener des calculs pour retrouver les schèmes de régulation dans les motifs reconnus en

interaction avec l'utilisateur. C'est l'utilisateur qui a l'initiative de guider la découverte de connaissance en cherchant à reformuler les interactions à un niveau d'abstraction rejoignant la manière dont il décrit sa propre activité. Depuis plus de 10 ans maintenant, l'étude du potentiel des *traces modélisées* est menée dans différents domaines, avec des fonctions d'assistance à l'appropriation des dispositifs et de leurs régulations (Champin et al., 2013).

Pour poser le problème de l'explicabilité, nous rappellerons comment elle se décline dans le domaine de l'informatique à l'ère de ce que l'on appelle couramment les *intelligences artificielles*, que nous situerons rapidement. Le problème posé, nous exposerons l'approche que nous adoptons dans nos recherches visant à mettre en intelligence l'utilisateur et son environnement informatique par des dispositifs menant à établir une *congruence* d'interprétations. En d'autres termes, nous étudions comment ajuster la compréhension de l'utilisateur et la sémantique des régulations intégrées dans le dispositif technique numérique. Après avoir présenté les modèles et les outils que nous développons pour les traces modélisées, nous illustrerons leur mobilisation sur plusieurs cas d'usage où les fonctions *d'explicabilité*, telles que nous les avons définies, ont été conçues, mises en œuvre et exploitées : assistance à la construction interactive d'un processus d'analyse de traces ; explicitation des processus d'analyse de traces pour leur appropriation en réutilisation ; exploration interactive de traces avec assistance à leur modélisation ; appropriation des fonctions de traçage et négociation de la construction d'indicateurs d'apprentissage. Nous conclurons cet article de positionnement en montrant le potentiel de cette approche et en traçant les perspectives ouvertes aussi bien théoriques que techniques que nous souhaitons discuter dans la communauté de *l'explicabilité* telle qu'elle commence à se former.

2 Intelligence, informatique, intelligence artificielle

C'est avec la conférence de Dartmouth en 1956, et la création du terme « Intelligence Artificielle », que la notion d'intelligence est appliquée à l'informatique. Il s'agissait alors de caractériser des dispositifs de calcul qui puissent émuler voire dépasser les fonctions cognitives, par exemple [1] : les fonctions exécutives, les fonctions visuo-spatiales, les gnosies, le langage, la mémoire, les praxies, et ceci en cherchant à défier la vitesse de traitement de l'information dans le cerveau.

L'examen de ces fonctions montre qu'elles ne sont pas indépendantes. Pourtant, elles ont été et sont encore souvent étudiées séparément, comme si elles n'avaient pas d'interactions les unes avec les autres dans leur fonctionnement. Au départ, la question principale était de découvrir ce que le cerveau (essentiellement) effectue comme traitement pour réaliser ces fonctions cognitives. Les fonctions exécutives ont été étudiées comme des formes de raisonnement formels ayant pour objet de déterminer la meilleure solution à un problème. L'hypothèse était faite que les problèmes étaient formalisables et que leur résolution était sans doute calculable (Newell et al., 1959). Les interactions fournissent des flux d'information entrant et sortant : des données à traiter et produisant d'autres données. Les fonctions cognitives autres que le raisonnement sont à l'évidence indissociables des possibilités d'interaction mêmes. Pour les étudier, elles sont souvent sorties de leur contexte corporel vivant en les considérant dans des contextes de corps artificiels (robots). Les applications de ces recherches consistent à pouvoir déléguer

1. Liste issue de `https://aqnp.ca/la-neuropsychologie/les-fonctions-cognitives/`

telle ou telle fonction cognitive humaine à un dispositif de calcul qui pourrait assurer cette fonction au moins aussi bien que l'humain. Une application, un dispositif technique numérique réalisé selon ces principes, sera nommé *intelligence artificielle* dès l'instant où une fonction cognitive humaine lui est déléguée. Les applications de recommandation, les systèmes de décision, les systèmes de diagnostic, de pronostic, de planification, d'organisation, de traduction,... sont légion. L'essentiel de l'informatique en interaction avec les humains pourrait alors relever de cette définition.

La question de la régulation [2] (Simondon, 1989) embarquée dans ces fonctions se pose : quelle est la sémantique de régulation exprimée *via* les règles codées ? Deux possibilités : les règles sont formellement décrites, en tant que données, pour être traitées par des algorithmes d'inférence ; les règles sont codées directement dans la fonction. Dans le premier cas, leur explicitation est possible par construction, à l'image du système expert pionnier Mycin (Buchanan et Shortlife, 1984). Dans le second cas, les règles codées peuvent être représentées de manière explicite dans un langage de programmation, ou cristallisées dans la structure même du dispositif, par exemple des réseaux neuronaux. Les règles sont associées aux informations, comme mode d'emploi. Cette association constitue une *connaissance* : le dispositif déclenchant la fonction sait quoi faire d'une information disponible. La mise en œuvre de ces connaissances réalise une sorte d'intelligence artificielle. L'apprentissage et la gestion de ces règles sont étudiées en tant qu'ingénierie des connaissances (Newell, 1982). Dans le cas des règles explicites exploitées par un moteur de raisonnement, l'ingénierie des connaissances propose des processus de collecte, représentation, gestion et exploitation sous des formes explicites. Le Web Sémantique s'inscrit dans cette catégorie. Dans le cas du codage par un programmeur directement dans l'algorithme, seul le code est explicite. Dans le cas de règles apprises par des dispositifs d'apprentissage automatique, on distinguera les approches symboliques, des approches non symboliques (Cornuéjols et al., 2018). Dans les deux cas, il est toujours très difficile de connaître la justification des règles, car le processus de collecte des données d'apprentissage n'est pas embarqué avec son résultat.

L'*explicabilité* des dispositifs techniques numériques est devenu un enjeu de société, à tel point que lorsque leur fonctionnement est régi par des règles issues des algorithmes d'apprentissage profond, elle donne lieu à des débats animés. [3] Dans cet article, nous considérons qu'un dispositif est explicable *lorsqu'il réunit les conditions pour permettre à l'utilisateur de s'approprier la sémantique des régulations à l'œuvre dans ce dispositif.* Cette définition impose au dispositif de pouvoir exploiter les descriptions explicites des règles disponibles dans l'environnement numérique, mais aussi de permettre à l'utilisateur d'explorer, à son initiative, le dispositif en fonctionnement. L'explicabilité ne dépend donc pas uniquement de l'explicitation mais aussi de la capacité à la projeter dans l'expérience utilisateur. La section suivante s'attache à définir plus précisément cette notion de « en intelligence ».

2. Tout objet technique est conçu avec des règles d'usage que le *technicien* doit comprendre pour le maîtriser. Si la règle d'usage d'un marteau peut se deviner en le manipulant, la règle d'usage d'un dispositif numérique se révèle bien plus difficilement, d'autant que le résultat de l'usage n'est pas observable directement

3. L'alerte dans un journal de vulgarisation scientifique `https://www.sciencesetavenir.fr/high-tech/intelligence-artificielle/intelligence-artificielle-l-explicabilite-talon-d-achille-du-deep-learning_124903` ; la théorisation de la question en relation avec la capacité à respecter la loi (RGPD par exemple) : `https://perso.math.univ-toulouse.fr/mllaw/home/statisticien/explicabilite-des-decisions-algorithmiques/`

3 Interagir en intelligence : congruence des interprétations

Nous avons vu comment des intelligences artificielles (IA dans la suite) pouvaient se mettre en place avec des capacités d'explicitation variables selon la technique utilisée. Dans le scénario d'usage d'une IA, l'explicabilité est dépendante de la conception de cette IA : soit la conception a prévu des fonctions d'explication exploitant les capacités intrinsèques d'explicitation de la technique utilisée, pour fournir les informations nécessaires (informations mobilisées, règles formelles déclenchées,...). L'utilisateur peut ainsi être informé *via* la documentation produite avec une possible mise en contexte – par exemple en fournissant une visualisation du graphe de relations sémantiques mobilisé pour produire un résultat (Hasan, 2014) ; soit ce n'est pas prévu, et l'utilisateur ne peut que tenter de deviner les régulations à l'œuvre en testant l'application avec des informations différentes, et par induction les imaginer. Quand il n'y a pas de substrat explicite (cas des réseaux neuronaux appris), des mécanismes dédiés peuvent être proposés pour re-construire du sens a posteriori à partir des régularités découvertes par la machine (Olah et al., 2018).

Mais quels que soient les efforts de conception pour rendre les dispositifs ainsi explicables, il demeure un problème de congruence [4] entre les constructions élaborées par le dispositif et celles compréhensibles par l'utilisateur. Cette congruence pose un problème dans les deux sens : difficulté pour l'utilisateur de comprendre une explication produite par le dispositif, difficulté pour le système d'adopter le registre [5] habituel d'expression de l'utilisateur (par exemple une expression en langue naturelle) pour son propre registre formel (par exemple une requête en langage informatique).

De nécessaires reformulations s'imposent pour passer d'un registre à l'autre, jusqu'à obtenir un effet de couplage entre le dispositif et l'utilisateur. La notion de *couplage* renvoie à l'idée qu'après avoir compris comment fonctionnait un système, il n'est plus besoin de réfléchir pour l'utiliser : on le fait *sans y penser*.

Comment alors construire des dispositifs ayant cette capacité d'explicabilité ? L'explicabilité étant une condition nécessaire (bien que non suffisante) pour définir des dispositifs techniques que l'utilisateur puisse s'approprier, nous reprenons et précisons ici les propriétés à donner à un dispositif pour lui conférer cette capacité d'explicabilité : 1) Caractère explicite des règles à l'œuvre dans les fonctions activées lors de l'utilisation d'un dispositif. Une fonction est explicite lors de son activation si elle dispose d'un mécanisme de description symbolique des règles à l'œuvre [6] ; 2) Propriété d'explicabilité des régulations à l'œuvre dans le contexte d'usage d'un utilisateur. Une fonction est explicable si l'utilisateur peut interagir avec un assistant d'explication permettant de mettre en relation l'expérience utilisateur liée au contexte d'usage et les informations explicitées disponibles.

Pour rendre *explicable* un dispositif technique numérique, nous proposons d'utiliser les traces d'utilisation comme information complémentaire pour co-construire les explications en interagissant avec l'utilisateur. Les traces d'utilisation sont articulables avec les explicitations

4. Dans ce texte, nous utilisons le terme congruence pour indiquer la possibilité d'alignement entre les interprétations (sémantique) faites d'une expression symbolique partagée (sémiotique).

5. Le registre d'expression est constitué de l'ensemble des éléments disponibles pour exprimer une information sous une forme symbolique explicite correspondant à la sémantique recherchée.

6. Cette propriété est très utile, mais pas garantie car dépendant des efforts de conception pour développer une facette explicitation de fonction. C'est objectivement difficile avec les outils de développement classique (applications compilées), c'est possible lorsque l'application exploite un moteur d'inférence associé aux connaissances représentées explicitement, c'est aujourd'hui presque impossible pour les fonctions représentées par un réseau neuronal appris.

disponibles dans l'environnement. Le rapprochement entre le point de vue utilisateur et le point de vue concepteur relève de la mise en congruence d'interprétations sur des formes sémiotiques. C'est la *congruence des interprétations* qui autorise le travail *en intelligence.*

4 Les traces *modélisées* comme conteneur de connaissances

L'équipe TWEAK du laboratoire LIRIS a proposé un méta-modèle pour la construction de systèmes à base de traces, des systèmes exploitant les connaissances présentes dans les traces d'interaction entre l'utilisateur et le système (Champin et al., 2013). La notion centrale de ce méta-modèle est celle de m-trace (*modeled trace*), définie comme une liste d'éléments observés appelés obsels (*observed elements*). Chaque obsel est décrit par un type, un ensemble d'attributs, et deux estampilles temporelles début et fin, délimitant l'intervalle durant lequel cet obsel a pu être observé [7]. Chaque m-trace est associée à un modèle de traces, qui spécifie les types d'obsels que la trace peut contenir, ainsi que les attributs de chaque type d'obsels. Ainsi, le modèle de traces permet d'expliciter la structure et la sémantique sous-jacente d'une m-trace. Cette connaissance est capitalisable, puisque plusieurs m-traces décrivant des activités similaires peuvent faire référence au même modèle. On peut par exemple imaginer un modèle de traces décrivant les traces d'interaction issues d'une plateforme d'enseignement à distance qui propose des cours et des exercices. Il pourrait ainsi y avoir trois types d'obsels : la *consultation d'un cours*, la *réponse à un exercice*, la *demande d'une aide* pendant un exercice. Ces trois types d'obsels partageront un attribut *identifiant l'apprenant*. Le type *consultation d'un cours* pourra avoir un attribut *identifiant le cours*. Le type *réponse à un exercice* aura un attribut indiquant l'*identifiant de l'exercice*, un attribut précisant le *type d'exercice* (QCM, QROC, appariement, etc.), un attribut indiquant à quel *cours* il se rapporte, un attribut décrivant son *niveau de difficulté*, et un attribut indiquant le *feedback* fourni par le système (par exemple réponse correcte ou incorrecte). Les m-traces sont stockées et traitées par un Système de Gestion de Traces Modélisées (SGTM). Un SGTM contient deux types de m-traces : les m-traces premières qui contiennent des obsels collectés directement depuis les applications ; et les m-traces transformées, dont les obsels sont calculés à partir du contenu d'une (ou plusieurs) m-trace(s) source(s), qui peuvent être indifféremment premières ou transformées. Les transformations sont notamment utilisées pour « élever » la description de l'activité d'un modèle bas niveau (focalisé sur les interactions *atomiques* [8]) vers un modèle de plus haut niveau (décrivant des actions ou des tâches plus abstraites).

kTBS [9] (kernel for Trace-Based Systems) est une implantation de référence open-source de SGTM. Il utilise le modèle de données RDF qui offre la flexibilité nécessaire pour représenter les m-traces selon divers modèles de traces. kTBS fournit un ensemble d'opérateurs de transformation, depuis de simples filtres jusqu'à des ré-écritures complexes spécifiées en SPARQL. Il permet également la définition d'opérateurs personnalisés. Techniquement, kTBS offre une API REST qui permet de créer, consulter et modifier les m-traces et leurs modèles.

7. Il est toujours possible que début et fin aient la même valeur, pour représenter des événements instantanés.
8. Une interaction est atomique lorsqu'elle n'est pas décrite en sous-interactions ; elle peut toutefois concerner une interaction décrite à un niveau très abstrait.
9. `http://tbs-platform.org/tbs/doku.php/tools:ktbs`

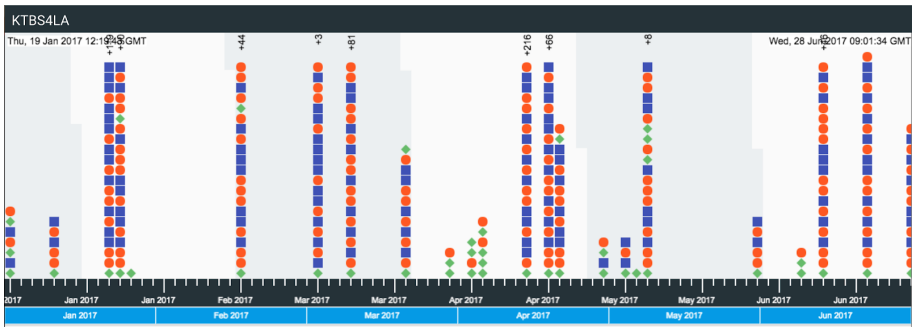

FIG. 1 – *Visualisation sous forme de timeline de l'activité des apprenants. Chaque obsel est représenté par un symbole dépendant des critères définis par l'utilisateur analyste (en fonction de son type et de ses valeurs d'attributs).*

5 L'explicabilité par les traces : illustrations

Permettre à un analyste de construire de manière interactive un processus d'analyse de traces. La plateforme kTBS présentée en section 4 est destinée à l'analyse de traces d'interaction. Dans le contexte des outils numériques pour l'éducation ou la formation, l'analyse des interactions entre les apprenants et l'environnement d'apprentissage (Learning Analytics) est nécessaire pour suivre la progression des apprenants, leur apporter un soutien, et mieux comprendre les mécanismes d'apprentissage. Les utilisateurs ayant besoin d'analyser les traces ne sont en général pas des spécialistes des technologies du web sémantique et ne peuvent pas exprimer des requêtes en SPARQL. Nous avons donc conçu la plateforme d'analyse de traces kTBS4LA [10] (kTBS for Learning Analytics), qui permet à un utilisateur de définir un modèle des traces qu'il souhaite analyser, de visualiser ces traces pour les explorer, d'effectuer des transformations de traces et de calculer des indicateurs sur l'activité des apprenants. Les spécificités de cet outil sont l'importance donnée à la temporalité des traces, une représentation explicite de la sémantique des interactions tracées, et la possibilité pour l'utilisateur de manipuler directement les données qu'il analyse. Pour analyser les traces issues d'un environnement d'apprentissage, l'analyste commence en effet par décrire ces traces en définissant leur modèle ; il donne ainsi du sens aux interactions tracées. Ce modèle de traces est utilisé pour proposer à l'analyste une visualisation des traces dans laquelle les obsels de différents types sont représentés différemment (cf. Figure 1). L'utilisateur analyste peut définir des règles décrivant quelle forme et quelle couleur utiliser pour représenter un obsel vérifiant certaines contraintes portant sur son type ou ses attributs. Par exemple, il pourra choisir de filtrer les données en faisant disparaître les obsels de type *consultation d'un cours*, de faire apparaître comme un losange vert les obsels de type *demande d'aide*, comme un carré bleu les obsels de type *réponse à un exercice* dont le *niveau de difficulté* est supérieur à un seuil et comme des ronds rouges ceux dont le *niveau de difficulté* est inférieur à ce seuil. Cet exemple de visualisation peut servir à l'utilisateur pour voir si les apprenants demandent davantage d'aide dans les exercices difficiles que dans les exercices plus faciles. L'ensemble des règles de visualisation ainsi définies par l'utilisateur sur une trace donnée constitue une feuille de style qui peut être

10. http ://tbs-platform.org/tbs/doku.php/tools :ktbs4la

enregistrée pour être ré-utilisée sur une autre trace ayant le même modèle. Différentes feuilles de style peuvent être utilisées pour visualiser les traces, pour répondre à différents besoins d'analyse. Une feuille de style peut également servir à créer un opérateur de transformation qui permettra de créer de nouvelles traces représentant un nouveau point de vue sur l'activité des apprenants. Cet opérateur de transformation pourra être appliqué à un ensemble de traces ayant le même modèle, même s'il a été défini initialement sur une trace donnée. Si l'on reprend l'exemple ci-dessus, les traces transformées n'auront plus de type d'obsels *consultation d'un cours* ni *réponse à un exercice*, et auront deux nouveaux types d'obsels : *réponse à un exercice facile* et *réponse à un exercice difficile*. La plupart du temps, les analyses sur les données d'apprentissage sont effectuées par des informaticiens et restituées aux équipes pédagogiques ou aux chercheurs sous forme de tableaux de bord qui ne correspondent pas toujours aux besoins des utilisateurs, ou à l'évolution de ces besoins (lorsque que les utilisateurs savent exprimer leurs besoins), ce qui nécessite de nouvelles analyses et la conception de nouveaux tableaux de bord. Avec kTBS4LA, l'analyste manipule directement les traces et peut construire de manière dynamique et interactive le processus d'analyse répondant à ses besoins. En permettant à l'utilisateur de donner une sémantique explicite aux traces, en lui permettant de construire lui-même les visualisations et donc les interprétations des données dont il a besoin, le mécanisme d'analyse est explicite et ses résultats sont explicables. On peut dans ce cas parler d'explicabilité *by design*.

Documenter des processus d'analyse de traces pour faciliter leur réutilisation. Dans la communauté des Learning Analytics, la question de la capitalisation des analyses est posée. D'un point de vue informatique, ces analyses sont concrétisées par des processus d'analyse de traces composés d'une succession ordonnée d'opérations, implantées dans un outil d'analyse, appliquées sur des traces d'apprentissage. Ces processus d'analyse sont soumis à des contraintes qui sont soit liées aux contextes d'apprentissage, soit aux spécificités techniques des données et des outils d'analyses. Ces contraintes rendent le partage, la réutilisation et l'adaptation des processus complexes voire peu pertinents (Clow, 2012). Pour faciliter la capitalisation des processus d'analyse de traces, nous nous appuyons sur une approche narrative visant à structurer sémantiquement un processus d'analyse de traces d'apprentissage ainsi que les informations associées, décrivant le contexte d'apprentissage, caractérisant les traces, et justifiant les choix d'analyse effectués (Lebis et al., 2018). Cette démarche, fondée sur une ontologie, vise à rendre les analyses compréhensibles, la compréhensibilité étant l'une des propriétés nécessaires à la capitalisation. Une analyse est compréhensible si les différents aspects de l'analyse sont appréhendables par les acteurs concernés. Pour cela, il faut fournir des informations techniques, mais aussi des informations conceptuelles, comme les objectifs de l'analyse, les théories scientifiques utilisées, ou encore les choix d'implantation. Nous proposons ainsi de décrire un processus d'analyse par une succession d'étapes dans lesquelles des opérateurs d'analyse sont utilisés. Un *opérateur narré* représente le concept d'opération commun à des opérateurs similaires, lorsque sa sémantique est non ambigüe, et lorsque différentes informations sont intégrées et structurées. Nous définissons le comportement de ces opérateurs sur les variables à l'aide de différents *patrons*, eux aussi sémantiquement définis. L'analyste peut associer à l'ensemble du processus d'analyse, mais également à chaque étape de ce processus, des éléments narratifs, fondés sur un vocabulaire partagé, et permettant d'intégrer de manière structurée des informations liées au contexte de l'analyse et aux choix effectués. Cette explicitation par l'analyste de différents éléments décrivant une analyse représente un effort

conséquent. Cependant, le fait d'utiliser une ontologie et un vocabulaire contrôlé permet la congruence d'interprétations entre le système et l'utilisateur, et permet ainsi au système de rendre des services à la communauté des utilisateurs analystes. En effet, si un utilisateur exprime un besoin d'analyse en utilisant le vocabulaire contrôlé, le système peut retrouver des processus d'analyse existants et répondant au même besoin. Il peut également montrer à l'utilisateur dans quelle mesure ces processus existants ont été utilisés dans des contextes similaires au contexte du besoin exprimé. Il peut enfin lui expliquer quelle adaptation effectuer sur un processus d'analyse existant pour faciliter son application à un contexte similaire.

Exploration de traces pour la découverte interactive de connaissances. Transmute (Fuchs et Cordier, 2018) est une approche de découverte interactive de connaissances à partir de traces qui s'appuie sur la fouille de données pour mettre en évidence des régularités dans les traces sous la forme de sous-séquences d'événements appelées épisodes séquentiels. L'utilisateur, expert du domaine d'application, est impliqué pour sélectionner des épisodes *intéressants* et les interpréter pour construire un modèle du phénomène étudié. Les obsels composant la trace sont associés à des représentations choisies par l'utilisateur qui font sens pour lui, des symboles graphiques par exemple, faisant référence à des concepts connus de l'utilisateur (figure 2). Les épisodes sont représentés de la même façon et localisés dans la trace pour améliorer leur compréhension en contexte. Le système à base de traces fournit des possibilités de transformations dont la réécriture qui crée, à partir des épisodes sélectionnés, une nouvelle trace d'un niveau plus abstrait associé à son modèle. Le modèle du phénomène étudié qui a été construit à l'issue de l'interprétation est explicable car il est possible de naviguer vers ses éléments constitutifs dans la trace dont il est issu, et ceci de manière récursive.

FIG. 2 – *Extrait de l'interface de Transmute avec des traces issues d'un jeu sérieux.*

KATIE [11] (Fuchs, 2017) est une approche d'acquisition de connaissances qui vise à assister, en interaction avec l'utilisateur, le processus de modélisation et d'intégration des traces dans un système à base de traces, en détectant et corrigeant les erreurs résiduelles dans les données (données bruitées ou manquantes). À partir d'une trace fournie sous la forme d'un jeu de données brutes, KATIE extrait un modèle de trace [12] qui est proposé à l'utilisateur. Lorsque le modèle proposé ne correspond pas aux connaissances que l'utilisateur possède sur les données, ce dernier peut exprimer ses connaissances sous la forme de contraintes sur les données. Les

11. Knowledge Acquisition from Traces with Interactive Exploration
12. Le terme schéma de classes est souvent utilisé, quoique non équivalent. Ici il s'agit d'une hiérarchie des types d'obsels qui correspond à un modèle de connaissances de la trace.

données discordantes par rapport à ces contraintes sont extraites et montrées à l'utilisateur qui décide des actions à mener pour corriger les éventuelles erreurs : suppression de données bruitées ou ajout de données manquantes. Une fois les données modifiées en conséquence, KATIE réitère la construction du modèle de trace jusqu'à ce qu'un consensus soit obtenu avec l'utilisateur. Ce dernier peut alors interpréter les différents concepts proposés. Finalement, le modèle de trace est créé dans le système à base de traces et la trace y est ensuite enregistrée conformément à ce modèle. La génération du modèle de trace s'appuie sur l'analyse de concepts formels (Ganter et Wille, 2012) et les contraintes de l'utilisateur sont exprimées à l'aide d'implications de la forme : *tous les obsels qui possèdent l'attribut x possèdent également l'attribut y*. Le processus explore itérativement et alternativement l'*intent* et l'*extent* pour expliquer et rectifier les désaccords entre les connaissances de l'utilisateur et le modèle sous-jacent aux traces. Ces approches donnent à l'utilisateur un rôle central dans la construction de connaissances et l'explicabilité y est une caractéristique recherchée. L'utilisateur peut lui-même choisir la représentation visuelle des éléments de la trace, et c'est *via* leur manipulation interactive, que les règles sous-jacentes à la découverte de connaissances sont explicitées. Les modèles de connaissances obtenus après interprétation sont ainsi sémantiquement et explicitement reliés aux données et informations dont ils sont issus.

S'approprier ses traces d'interaction et négocier les indicateurs d'apprentissage associés : le prototype Trace-Me. Dans le cadre de la mission COAT [13] menée par le CNRS en 2014, une étude a été réalisée pour associer à un dispositif d'apprentissage MOOC, un dispositif d'assistance aux utilisateurs apprenants pour l'appropriation et le contrôle des fonctions de traçage, et d'élaboration d'indicateurs à partir des traces d'interaction. Cette étude a permis d'expérimenter une manière d'implanter deux « bonnes propriétés » dans le dispositif pour l'appropriation par l'utilisateur des régulations à l'œuvre, ce que nous défendons ici comme définitoire de l'explicabilité :

1. Appropriation par l'utilisateur du processus de traçage
   - démarrage, suspension, reprise et arrêt de traçage sous son contrôle ;
   - choix de ce qui doit être tracé sur son poste pour une activité donnée : les apprenants choisissaient souvent de tracer l'accès à des ressources autres que celles offertes par la plateforme MOOC ;
   - stockage dans un environnement privé ;
   - visualisation interactive en temps réel, avec un effet de réflexivité de l'activité dans le temps et dans l'espace :
     - de ses interactions vécues dont il a l'expérience action-perception ;
     - des actions de la plateforme, non vécues par lui mais révélées en proximité visuelle de ses propres interactions et fournissant des clés du fonctionnement interne ;
   - mise en forme configurable : symboles, couleurs, champs textes ...
2. Appropriation des logiques de régulation par des fonctionnalités d'exploration et d'intervention sur la sémantique des traces synthétisées dans les indicateurs :
   - accès à un outil d'analyse des traces selon des modèles configurables,
   - accès à un outil de gestion d'indicateurs partagés dans un *store*, permettant d'accéder aux modèles des indicateurs, et d'y déposer les siens propres, développés avec l'outil de conception utilisé par les concepteurs enseignants (backoffice). Les indicateurs sont

13. Connaissance Ouverte A Tou.te.s `https://projet.liris.cnrs.fr/coatcnrs/wiki/doku.php`

affichés dans l'environnement d'apprentissage des utilisateurs apprenants (frontoffice). L'apprenant a accès au backoffice et au frontoffice.

Ce dispositif a été expérimenté lors de formations MOOC et pendant une école d'été du CNRS. Les plateformes de MOOC sont en réalité peu utilisées comme espace de travail collectif. Les apprenants utilisent massivement les outils génériques du Web qu'ils connaissent pour travailler ensemble. L'articulation entre plateforme et autres outils pour une même activité de l'utilisateur n'est pas assurée en général. Trace-me offre la possibilité d'observer son activité d'apprentissage qu'elle soit médiée par la plateform MOOC ou non. L'utilisateur maîtrise son traçage propre et peut intervenir dans la régulation en proposant d'adapter ou de créer de nouveaux indicateurs, en discussion avec les utilisateurs concepteurs ou non.

Dans cet exemple, on voit que le processus d'explication exige une explicitation de la régulation de l'apprentissage (traces et modèles d'indicateurs) mais aussi des fonctions d'exploration et de reformulation par l'utilisateur lui-même dans un engagement dans l'effort d'explication. Des prototypes similaires sont décrits par Cram et al. (2007, 2008).

6 Discussion et perspectives

Dans son analyse de l'opacité des systèmes d'IA, Burrell (2016) en identifie trois formes : l'opacité liée à un effort délibéré de garder *secret* le fonctionnement du dispositif, celle liée au *manque de compréhension* par les utilisateurs, et celle liée à la *complexité* intrinsèque du dispositif. Cette dernière forme n'est pas limitée aux systèmes d'IA numérique, car elle ne provient pas uniquement du manque d'explicitation : la combinatoire d'exécution d'un algorithme non-trivial en rend son appréhension difficile, y compris pour un spécialiste.

L'opacité liée au secret sort quelque peu du cadre de cet article, puisque notre approche suppose que les capacités d'explicabilité sont recherchées par les concepteurs du dispositif. On peut cependant noter qu'un effort de rétro-conception, s'appuyant sur des traces d'intectactions collectées *via* une observation extérieure au dispositif (Ginon et al., 2013), reste possible pour rendre un tel système explicable « malgré lui ».

L'opacité liée au manque de compréhension est la raison pour laquelle nous considérons insuffisante (bien que nécessaire) l'explicitation des régulations à l'œuvre dans le dispositif. Cette dernière doit s'accompagner de mécanismes interactifs de négociation de sens, pour mettre en congruence l'interprétation personnelle de l'utilisateur avec l'intérprétation canonique prescrite par les concepteurs du système. En tant qu'elles capturent une « expérience » partagée entre l'humain et le dispositif, les traces d'interaction constituent un médium de choix pour alimenter cette négociation. Elles sont éminemment *ambivalentes*, en ce sens qu'elles peuvent avoir une signification différente pour le système et pour l'utilisateur, mais aussi pour différents utilisateurs (par exemple un apprenant et un enseignant, dans le contexte d'une application éducative). Nous avons proposé (Champin, 2017) un cadre théorique pour apréhender cette ambivalence et formaliser les congruences entre les différences interprétions pouvant coexister dans ce cadre. Sa mise en pratique concrète reste cependant à évaluer dans des travaux futurs.

Enfin, la notion de trace transformée, centrale dans le modèle proposé en section 4, vise précisément à répondre à l'opacité liée à la complexité. Elles permettent en effet, d'une part, de reformuler des traces de « bas niveau » en traces plus abstraites et synthétiques, et d'autre part, d'expliquer les obsels transformés par leur lien généalogique avec les obsels sous-jacents. Bien

sûr, ces transformations elles mêmes ont leur part de complexité. Il convient donc de veiller à ce que l'explication ne soit pas plus opaque que le système qu'elle cherche à expliquer.

7 Conclusion

Cet article a été l'occasion de rappeler l'origine de l'ambition des recherches en intelligence artificielle, cherchant à étudier les fonctions cognitives fondamentales, pour les comprendre et en tirer profit en concevant des dispositifs techniques qui s'en inspirent pour être efficaces dans le traitement de tâches complexes. Certains systèmes se voulaient *explicables* par construction puisque mimant le raisonnement humain pour la résolution de problèmes, avec des heuristiques explicites et un effort à montrer la rationalité du raisonnement par sa trace symbolique et vérifiable. D'autres systèmes étudiaient les structures qui se construisent à partir d'un apprentissage *automatique* sans formulation d'un raisonnement explicite. Nous soutenons que l'explicabilité nécessite l'explicite mais le dépasse pour exiger une capacité à rapprocher les interprétations que l'utilisateur peut élaborer en situation d'utilisation d'un dispositif technique numérique et les régulations à l'œuvre telles qu'elles déterminent le comportement dudit dispositif. Nous parlons de *congruences d'interprétations*. Nous proposons d'utiliser les traces d'interaction comme support commun à l'utilisateur et au dispositif technique numérique pour l'exploration des régulations s'exprimant au travers des motifs d'interaction, à différents niveaux d'abstraction. La transformation d'une trace peu explicite en une trace interprétée nécessite une transformation explicite qui décrit d'une certaine manière les règles d'interprétation correspondant à telle ou telle régulation à l'œuvre dans cette séquence d'interactions. Lorsque l'utilisateur n'est pas surpris par le comportement du dispositif technique numérique, c'est qu'il l'a compris, qu'il est maintenant le technicien de la fonction concernée au sens établi par Simondon (1989) pour exprimer cette maîtrise. Les illustrations montrent naturellement des cas d'usage des traces en situation réflexive pour les utilisateurs, mais aussi comment les concepteurs s'emparent de la notion de trace modélisée et des outils théoriques et techniques développés autour de cette notion pour préparer les conditions de l'explicabilité. Les perspectives montrent que le champ d'usage des traces modélisées s'élargit, se formalise et porte un potentiel fort pour les travaux à venir en matière d'explicabilité.

Références

Buchanan, J. M. et E. H. Shortlife (1984). *Rule Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project.* (Addison-Wesley ed.). Readin, MA.

Burrell, J. (2016). How the machine 'thinks' : Understanding opacity in machine learning algorithms. *Big Data & Society 3*(1), 12.

Champin, P.-A. (2017). *Empowering Ambivalence – Supporting multiple interpretations in knowledge-based systems*. HDR, Université Claude Bernard - Lyon I, Lyon, France.

Champin, P.-A., A. Mille, et Y. Prié (2013). Vers des traces numériques comme objets informatiques de premier niveau. *Intellectica* (59), 171–204.

Clow, D. (2012). The learning analytics cycle : closing the loop effectively. In *Proc. of 2nd International Conference on Learning Analytics and Knowledge*, pp. 134–138. ACM.

Cornuéjols, A., L. Miclet, et V. Barra (2018). *Apprentissage Artificiel. Deep learning, concepts et algorithmes (3rd Ed)* (Eyrolles ed.).

Cram, D., B. Fuchs, Y. Prié, et A. Mille (2008). An approach to user-centric context-aware assistance based on interaction traces. *MRC 2008, Modeling and Reasoning in Context*.

Cram, D., D. Jouvin, et A. Mille (2007). Visualizing Interaction Traces to improve Reflexivity in Synchronous Collaborative e-Learning Activities. In A. C. Limited (Ed.), *6th European Conference on e-Learning*, pp. 147–158.

Fuchs, B. (2017). Assister l'utilisateur à expliciter un modèle de trace avec l'analyse de concepts formels. In C. Roussey (Ed.), *IC 2017*, Caen, France, pp. 151–162.

Fuchs, B. et A. Cordier (2018). Interactive interpretation of serial episodes : experiments in musical analysis. In C. Faron-Zucker et C. Ghidini (Eds.), *EKAW-2018*, LNAI 11 313, Nancy, France, pp. 131–146. Springer.

Ganter, B. et R. Wille (2012). *Formal concept analysis : mathematical foundations*. Springer Science & Business Media.

Ginon, B., P.-A. Champin, et S. Jean-Daubias (2013). Collecting fine-grained use traces in any application without modifying it. In *workshop EXPPORT from the conference ICCBR*.

Hasan, R. (2014). *Predicting query performance and explaining results to assist Linked Data consumption*. Theses, Université Nice Sophia Antipolis.

Lebis, A., M. Lefevre, V. Luengo, et N. Guin (2018). Capitalisation of analysis processes : enabling reproducibility, openness and adaptability thanks to narration. In *Proc. of the 8th International Conference on Learning Analytics and Knowledge*, pp. 245–254. ACM.

Liquète, V., E. Delamotte, et F. Chapron (2012). L'éducation à l'information, aux tic et aux médias : le temps de la convergence ? *Études de communication 38*.

Mille, A. (2019). Vers des dispositifs techniques numériques orientés éthique ? *Intellectica* (70), 119–164.

Newell, A. (1982). The Knowledge Level. *Artificial Intelligence* (18), 87–127.

Newell, A., J. C. Shaw, et H. A. Simon (1959). Report on a general problem-solving program. In *Proceedings of the International Conference on Information Processing*, pp. 256–264.

Olah, C., A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, et A. Mordvintsev (2018). The Building Blocks of Interpretability. *Distill 3*(3).

Simondon, G. (1989). *Du mode d'existence des objets techniques*. Philosophie. Aubier.

Summary