Approche générique et adaptative de l'explication des maisons intelligentes

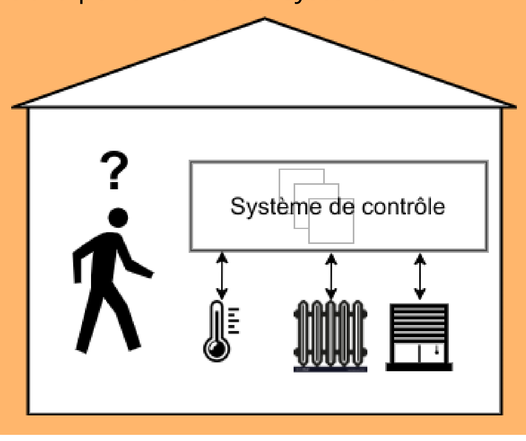
Étienne Houzé
Jean-Louis Dessalles
Ada Diaconescu
David Menga
Mathieu Schumman

Motivation

Vous vous trouvez dans une pièce, dont les volets se baissent soudainement. ***Pourquoi ?***

Dans le cadre d'un système contrôlant un environnement physique, comme une maison intelligent, il arrive que le système puisse prendre des décisions qui semblent **suprenantes** ou **non désirées** à l'utilisateur.

Avoir accès à une explication permettrait de rétablir la confiance dans le systeme et de réduire les interventions inutiles de l'utilisateur qui peuvent nuire à la performance du système !



Le système de contrôle peut Apparaître comme une boîte noire à l'utilisateur !

Difficultés

Expliquer les événements rares survenant dans un environnement physique n'est pas chose aisée. Plusieurs verrous expliquent le manque actuel de solutions:

- **Adaptation au changement**: Les systèmes tels que les maisons intelligentes sont appelés à évoluer au cours du temps: des équippements peuvent être ajoutés, retirés ou modifiés. Un système explicatif doit pouvoir prendre en compte ces changements sans perdre en efficacité.
- **Événement rares ou imprévus:** Les événements nécéssitant le plus une explication sont par nature suprenants ou rares. Dans ces situations, une simple association, comme le font les approches Machine Learning traditionnelles, semble vouée à l'échec.
- **Parler une "langue" commune**: Les différents équipements utilisent des grandeurs physiques qui leur sont propres, alors que l'utilisateur utilise un vocabulaire souple pour décrire ses observations du système.
- **Complexité des interactions**: Les nombreuses interactions entre equipements, la singularité de la configuraiton exacte de la maison, rendent inenvisageable le recours à une base de connaissance pré-établie, comme un connaissance ontologique basée sur des règles.

Quels mots ?

Les composants ont chacun leurs variables, leurs unités, qui peuvent différer et ne pas correspondre au ressenti de l'utilsateur. Notre approche propose donc d'identifier des **événements** depuis les flux de données enregistrées par les différents capteurs.

Ces événements sont classés, selon une arborescence de types. Ainsi, une température anormalement élevée sera catégorisée comme "événement", "événement de température" et "température elevée", et possèdera plusieurs attributs comme son moment d'apparition, sa durée, la température maximale enregistrée... La classe de l'événement permet de lui attribuer un prédicat, qui est ensuite utilisé pour décrire le système, à un instant donné : *hot(room), open(window), ...*

Afin d'encoder l'opinion de l'utilisateur ou du système sur ces prédicats, on leur attribue une **nécéssité**, un nombre dont le signe indique si l'a-priori est positif ou négatif, et la valeur absolue encode la force de cet a-priori. Par exemple, *cold(room): -20* indique que l'on ne veut pas, ou pense pas, qu'il fasse froid avec une intensité de 20 ; *open(window): 10* indique que l'on veut que la fenêtre soit ouverte, avec une intensité plus faible de 10. **[1]**

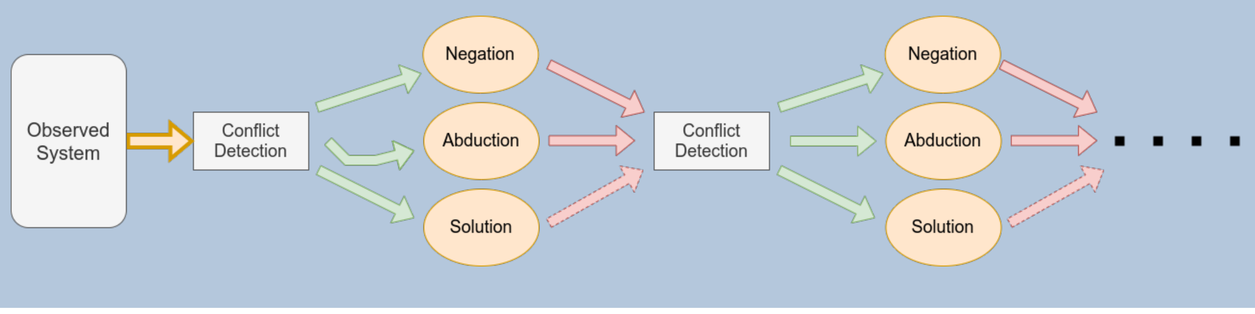
L'explication délibérative

Une demande d'explication naît d'un conflit initial entre deux situations incompatibles ; il s'agit souvent d'une conflit entre une observation et une croyance. Par exemple, on se demande pourquoi la fenêtre s'est ouverte car l'on s'attendait plutôt à ce qu'elle soit fermée.

De par cette nature conflictuelle, l'explication peut faire appel au raisonnement délibératif ou argumentatif. On se propose donc d'utiliser des méthodes de génération de dialogue argumentatif pour produire une explication. La trace d'un tel dialogue argumentatif pourra ensuite être proposé comme explication.

La procédure **CAN (Conflict-Abduction-Negation) [1]** propose une méthode séquentielle et minimale de générer une telle argumentation. Elle repose la détection de conflit, puis les propage à leurs possibles causes ou conséquences via un processus d'abduction (inférence des causes) ou une négation permettant de considérer l'état des choses inverse. Un conflit peut être résolu, en réalisant une action sur l'environnement ou en revoyant l'opinion associée.

Dans le contexte de la maison intelligente, la procédure CAN est adaptée afin de se conformer aux spécificités : la négation purement sémantique est remplacée par la simulation d'un contrefactuel, et les opérations d'abduction, ainsi que les connaissances sur l'état du système, sont partagées entre des **composants "explicatifs"** appairés aux composants du système de contrôle. Un composant central, nommé le **spotlight**, se charge de coordonner les subordonnés afin de composer leurs explications locales en un tout cohérent. **[2]**



Réduire la complexité

Comment trouver une méthode générique d'explication, malgré les différents contextes ? Une solution peut être de considérer la **complexité algorithmique**, qui peut être vue comme la plus courte manière de décrire un objet, une situation, un événement: $K(x)=min\{l(p), A(b)=x\}$ **[3]**

Dans le contexte de la maison intelligente, on peut évaluer l'importance d'un événement enregistré comme le plus court programme permettant de le retrouver au sein de la mémoire M à l'aide d'une « fonction de récupération » *f*.

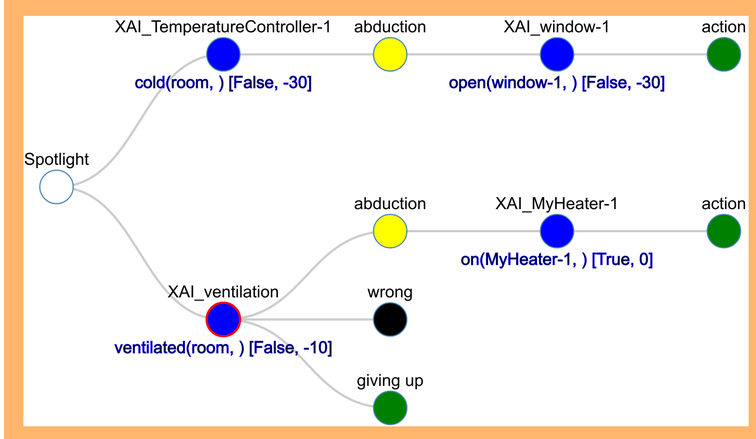
$$K(e) \leq min\{l(p), f(M, p)=e\}$$

Dans le cadre d'une explication, une possibilité pour proposer les hypothèses les plus pertinentes serait donc d'évaluer quels événements, parmi les enregistrements, conduisent à la description la plus simple pour l'observation questionnée. On peut borner la complexité de la relation $h \rightarrow e$ par la longueur du plus court programme permettant de retrouver *e* à partir de *h* et de la base de cas *CB :*

$$K(h \rightarrow e) \leq min\{l(p), f(CB, h, p)=e\}$$

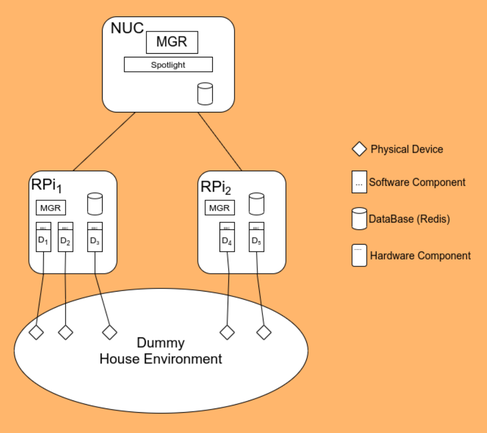
Implémentation

Les différents principes présentés ici ont été mis en application afin de réaliser une implémentation preuve de concept. Un interface web est disponible à l'adresse suivante https://explainableai.fr/, ainsi que quelques exemples simples commentés. Notemment, le système est capable de fournir une explication, sous forme d'un arbre, une situation ou la température est anormalement basse, du fait de l'aération de la pièce en ouvrant la fenêtre.



Le système explicatif, interrogé sur le froid, a commencé par donner la main au composant responsible de la température. Par un raisonnement d'abduction, celui-ci a propagé le conflit au composant responsible de la fenêtre. Étant donné qu'il lui est possible de directement fermer la fenêtre, il a donc demandé la simulation des conséquences de cette action. Il s'avère que fermer la fenêtre permet bien de faire remonter la température mais provoque également un nouveau conflit relevé par le composant responsable de la ventilation. Une abduction est tentée, mais est vite infirmée par une simulation. N'ayant plus d'autre option, le système doit donc vivre avec ce dernier conflit.

Cette première réalisation sur simulateur permet de montrer la viabilité de notre approche, mais de nombreux tests supplémentaires sont nécessaires. Nous travaillons donc actuellement sur une nouvelle version, bien plus complète, portant cette fois sur une maquette d'une maison intelligente. Cette seconde monture, prévue pour la fin de l'année 2021, montrera une implémenation sur des vrais composants, proche d'un produit embarqué. Le schéma ci-contre montre l'architecture envisagée : les équipements physiques seront connectés à des Raspberry Pi, eux-mêmes connecté à un NUC chargé de l'organisation générale. Les composants explicatifs seront déployés sur les RPi et le Spotlight sera intégré au NUC.



[1] : *A cognitive approach to relevant argument generation*, JL-Dessalles, in Principles and Practives of Multi-Agent Systems, 2016

[2] : A Decentralized Explanatory System for Intelligent Cyber-Physical Systems, E. Houzé, JL. Dessalles, A. Diaconescu, D. Menga, M. Schumann, in IntelliSys 2021

[3] : *An Introduction to Kolmogorov Complexity and its applications,* Li, Vitányi et al. 2008

Journée des doctorants – IP Paris – Juin 2021