



HAL
open science

Anti-patrons partiels pour l'identification des problèmes de contradiction sociale dans une ontologie

Mounira Harzallah

► **To cite this version:**

Mounira Harzallah. Anti-patrons partiels pour l'identification des problèmes de contradiction sociale dans une ontologie. IC2016: Ingénierie des Connaissances, Jun 2016, Montpellier, France. hal-01442728

HAL Id: hal-01442728

<https://hal.science/hal-01442728v1>

Submitted on 1 Feb 2017

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.

Anti-patrons partiels pour l'identification des problèmes de contradiction sociale dans une ontologie

Mounira Harzallah

LINA, Université de Nantes
mounira.harzallah@univ-nantes.fr

Résumé : Dans le contexte du web sémantique et des masses de données, les ontologies nécessaires à faire communiquer les objets du web seront probablement de grande taille et construites à partir des ressources diverses et hétérogènes. Leur validation sera plus que jamais primordiale. Dans cet article, nous mettons en exergue les liens entre deux types problèmes pouvant nuire à la qualité d'une ontologie : l'insatisfiabilité d'artefacts et le problème de contradiction sociale. Nous proposons des anti-patrons partiels et une heuristique pour aider à identifier les problèmes de contradiction sociale, tout en minimisant l'intervention humaine.

Mots clés : Validation d'ontologie, Insatisfiabilité, Problème de contradiction sociale, Anti-patron partiel.

1 Introduction

Dans le contexte du web sémantique et des masses de données reliées sur internet, les ontologies nécessaires à faire communiquer les objets du web seront probablement de grande taille et construites à partir des ressources diverses et hétérogènes. Leur validation sera plus que jamais primordiale et complexe. Dans nos travaux, nous considérons la validation comme un processus qui devrait être réalisé en parallèle avec celui d'acquisition/extraction des artefacts de l'ontologie (Harzallah et al. 2015). Il devrait être réalisé le plus tôt possible afin d'éviter la propagation des problèmes dans une ontologie et rendre complexe sa correction. En outre, nous considérons la validation comme un processus : 1) qui cherche une mauvaise qualité dans une ontologie et 2) qui propose ensuite une méthode pour améliorer cette qualité en identifiant et en enlevant les causes liées à cette mauvaise qualité. Des méthodes ont été proposées pour l'évaluation de la qualité d'une ontologie. Actuellement, il n'y a pas encore un consensus sur comment une ontologie doit être évaluée (Neuthaus & Vizedom, 2013). En plus, certains problèmes qui peuvent nuire à la qualité d'une ontologie sont peu traités et ne peuvent être identifiés que par un acteur social. Nous nous intéressons aux problèmes de contradiction sociale (i.e. une contradiction entre la perception qu'un acteur social a d'une ontologie et le contenu de cette ontologie), par exemple, dans une ontologie une relation de subsumption jugée fautive par un acteur social. Nous cherchons à identifier ces problèmes en les reliant aux problèmes d'insatisfiabilité qui sont identifiables d'une façon automatique.

Dans cet article, nous discutons des méthodes d'identification des problèmes pouvant affecter la qualité d'une ontologie. Ensuite, nous proposons des anti-patrons partiels de l'insatisfiabilité et une heuristique pour détecter des problèmes de contradiction sociale.

2 Méthodes d'identification des problèmes affectant la qualité d'une ontologie

Plusieurs termes sont utilisés pour évoquer la notion de problème affectant la qualité d'une ontologie : 1) les erreurs de taxonomie (Gomez-Perez et al. 2001) ou les erreurs structurelles (Buhmann et al. 2011), 2) les anomalies de conception ou défaillances (Baumeister & Seipel,

2005), 3) les anti-patrons (Roussey et al. 2010), (Buhmann et al. 2011), 4) les embûches ou les mauvaises pratiques (Poveda et al. 2012) et 5) les défauts logiques (Buhmann et al. 2011). Vu l'hétérogénéité et la diversité de ces termes et le recouvrement de leur définition, nous avons proposé une typologie des problèmes permettant de standardiser leur définition et d'aider à les identifier (Gherasim, 2013). Cette typologie est définie selon deux dimensions : aspect logique vs aspect social de l'ontologie et erreurs vs situations indésirables. Les erreurs sont des problèmes qui rendent une ontologie inutilisable et les situations indésirables sont ceux qui l'affectent sans empêcher son utilisation. Les problèmes logiques sont reliés à l'interprétation que donnent les machines à une ontologie sur la base de ses spécifications logiques. Les problèmes sociaux sont reliés à la perception/interprétation qu'un acteur social a d'une ontologie. Le terme « acteur social », emprunté à la terminologie utilisée dans le modèle de qualité des modèles conceptuels SEQUAL (Krogstie et al. 1995), mentionne une partie prenante (un individu ou un groupe d'individus) concernée par l'ontologie en question.

Nous avons formalisé la majorité des problèmes logiques en logiques de description en considérant les notions synthétisées dans Guarino et al. (2009) i.e. Interprétation (I), Modèles intentionnels (Intended Models) (IM), Langage (L), Ontologie (O), et les 2 relations : \models et \dashv (Gherasim, 2013). Nous formalisons certains problèmes sociaux, en rajoutant la notion de SAP_O (Social Actor Perception of Ontology) par analogie avec les notions de IM. Par exemple, $SAP_O \not\models \phi$ implique ϕ est fausse dans la perception d'un acteur social de O.

Plusieurs travaux se sont intéressés à l'étude des problèmes pouvant affecter une ontologie et les techniques de leur identification. Des raisonneurs (Pellet, FactC++, Racer) ont été développés pour identifier certains problèmes logiques, particulièrement les problèmes d'inconsistance, d'insatisfiabilité ou de redondance, mais sans déterminer leur origine. Des travaux relativement récents ont cherché à identifier et à présenter à l'utilisateur le plus petit ensemble d'axiomes menant à une insatisfiabilité (Wang et al. 2005) (Rodler et al. 2013). Pour déterminer la cause ayant la meilleure chance d'être la bonne et en minimisant le nombre de questions à poser à un utilisateur, Rodler et al. (2013) utilisent une méthode probabiliste prenant en compte les causes les plus fréquentes. Des nouvelles heuristiques cherchent à détecter des problèmes qui ne sont pas nécessairement détectés par des raisonneurs, par exemple des candidates à l'absence d'exclusivité (disjointness) (Quazi & Qadir, 2011). Roussey et al. (2010) ont défini des anti-patrons pour l'identification de certains problèmes d'insatisfiabilité ou des candidats à des problèmes d'ontologie inadaptée ou d'ontologie non minimale. L'identification d'autres problèmes logiques, par exemple l'incomplétude d'un point de vue logique, demande la génération de IM, ce qui n'est pas toujours évident à avoir. Concernant les problèmes sociaux, des approches récentes utilisent des systèmes de question/réponse pour les identifier (Pammer, 2010). Les questions portent souvent sur la totalité des artefacts d'une ontologie ce qui permet une validation complète mais qui repose sur l'utilisateur qui pourrait commettre des erreurs en répondant à des centaines de questions. Poveda et al. (2012) ont développé l'outil OOPS ! pour aider à détecter certains de ces problèmes, mais il n'est pas possible de comprendre clairement ce que cet outil peut détecter.

Dans la suite, nous présentons notre approche pour la définition d'anti-patrons partiels pour les problèmes d'insatisfiabilité et de contradiction sociale. Nos travaux complètent ceux portant sur l'identification des cas susceptibles de correspondre à des problèmes, par exemple, les travaux de Roussey et al. (2010) ou de Quazi & Qadir (2011) et ceux qui font intervenir l'utilisateur pour l'aide à l'identification des problèmes.

3 Formalisation des problèmes de contradiction sociale et d'insatisfiabilité

Dans une ontologie, certains problèmes peuvent être dus à ou impliquer l'existence d'un autre problème. L'identification des liens entre problèmes est un moyen pour bien optimiser le processus de validation d'une ontologie (Harzallah et al. 2015). Par exemple : un problème de redondance de relations de subsumption peut être dû à l'existence d'une relation de subsumption fautive, qui pourrait correspondre à un problème de contradiction sociale, et

l'analyse de cette redondance peut aider à cibler cette relation. Nous avons cherché à déterminer des liens entre des problèmes en se basant sur leur formalisation. En effet, il s'agit pour deux problèmes P1 et P2 et leur formule respective F(P1) et F(P2) de déterminer quelle information F nécessaire pour que F(P1) & F soit équivalente à F(P2). Des liens existent d'une façon intuitive entre les problèmes d'insatisfiabilité et les problèmes d'inconsistance logique, d'ontologie inadaptée ou de contradiction sociale. En effet, les problèmes d'insatisfiabilité représentent des artefacts dont l'interprétation est vide. Cependant, on définit rarement ce type d'artefacts. Ceci implique que probablement, on possède des instances pour l'artefact insatisfiable et en les rajoutant à l'ontologie on produit un problème d'inconsistance. Ce dernier cache souvent l'existence de quelque chose qui est fausse par rapport à des IM (problème d'ontologie inadaptée) ou par rapport à la perception d'un acteur social (problème de contradiction sociale). Par exemple, le problème d'insatisfiabilité P36 (Table 1) dans une ontologie O où $F(P36) = \{O \models Ci \sqsubseteq \forall R.Cj, Ci \sqsubseteq \exists R.Ck, Cj \sqcap Ck \equiv \perp\}$ peut impliquer l'existence :

- des problèmes d'inconsistance logique. En effet, le rajout de l'information $\{x \in Ci^I\}$ dans cette ontologie engendre une inconsistance (Ci^I étant une interprétation de Ci) ;
- des problèmes d'ontologie inadaptée. En effet, la possession de l'information : $IM \models Ci \sqsubseteq \exists R.\neg Cj$, $IM \models Ci \sqsubseteq \forall R.\neg Ck$ ou $IM \models \text{not}(Cj \sqcap Ck \equiv \perp)$ (IM étant un modèle intentionnel de L) implique l'existence d'un cas d'ontologie inadaptée dans O ;
- des problèmes de contradiction sociale. En effet, la possession de l'information : $SAP_O \models \text{not}(Ci \sqsubseteq \forall R.Cj)$, $SAP_O \models \text{not}(Ci \sqsubseteq \exists R.Ck)$ ou $SAP_O \models \text{not}(Cj \sqcap Ck \equiv \perp)$ implique l'existence d'un cas de contradiction sociale dans O.

Pour formaliser la relation de « un problème peut impliquer l'existence d'un autre problème » entre les problèmes d'insatisfiabilité et de contradiction sociale, nous avons, tout d'abord, utilisé notre typologie pour classifier 60 cas de problèmes traités dans la littérature (nous les avons numérotés de P1 à P60) : 33 cas (de P1 à P33) traités dans le catalogue de Poveda et al. (2012), 11 cas (de P34 à P44) considérés dans (Roussey et al. 2010), 10 cas (de P45 à P54) proposés dans (Wang et al. 2005) et présentés dans (Buhmann et al. 2011), 6 cas (de P55 à P60) proposés dans (Fahad et al. 2008). Cette classification a montré que la majorité de ces cas correspond à un des 5 types de problèmes suivants : Insatisfiabilité (12 cas), Non minimale (12 cas), Contradiction sociale (11 cas), Erreurs de conception (12 cas), et Incomplétude du point de vue social (9 cas). Nous nous sommes focalisés sur les cas d'insatisfiabilité et de contradiction sociale. Parmi les 12 cas d'insatisfiabilité étudiés, seulement 8 cas sont différents (lignes grisées dans la table 1).

TABLE 1 - Problèmes d'insatisfiabilité traités dans la littérature

Problèmes d'insatisfiabilité	Formalisation
P34-Anti-pattern AndisOR	$O \models C \sqsubseteq \exists R.(C_i \sqcap C_j), C_i \sqcap C_j \equiv \perp$
P36 - Anti-pattern UniversalExistence	$O \models C \sqsubseteq \forall R.C_i, C \sqsubseteq \exists R.C_j, C_i \sqcap C_j \equiv \perp$
P37 - Anti-pattern EquivalencelsDifference	$O \models C_i \equiv C_j, C_i \sqcap C_j \equiv \perp$
P45 - Partition error	$O \models C \sqsubseteq (C_i \sqcap C_j), C_i \sqcap C_j \equiv \perp$
P47 - Having both a class and its complement as super condition	$O \models C \sqsubseteq (C_i \sqcap \neg C_i)$
P48 - Having a super condition that is assumed to be disjoint	$O \models C \sqsubseteq \neg T$
P49 - Having a super condition that is an existential restriction that has a filler which is disjoint with the range of the restricted property	$O \models C \sqsubseteq \exists R.C_i, R.\text{Range} \equiv C_j, C_j \sqcap C_i \equiv \perp$
P50 - Having an universal restriction with Nothing as the filler and a must existing restriction along property relationships.	$O \models C \sqsubseteq \forall R.\perp, C \sqsubseteq \exists R.C_i, C_i \neq \perp$
P51 - Having more than allowed existential restrictions	$O \models C \sqsubseteq <2R.T, C \sqsubseteq \exists R.C_i, C \sqsubseteq \exists R.C_j, C_i \sqcap C_j \equiv \perp$
P52 - Having a super condition containing conflicting cardinality restrictions	$O \models C \sqsubseteq >nR.T, C \sqsubseteq <nR.T$
P53 - Inconsistence from other ressources	$O \models C \sqsubseteq \exists R.C_i, C_i \sqsubseteq C_j, C_j \equiv \perp$
P54 - Having a super condition that is an existential restriction where the domain of the restricted property is disjoint with it	$O \models C_i \sqsubseteq \exists R.T, R.\text{Domain} \equiv C_j, C_i \sqcap C_j \equiv \perp$

La table 2 comprend la formalisation des 11 cas de problème de contradiction sociale. Leur analyse a mis en évidence l'existence de 6 cas d'égalité (P05, P19, P25, P27, P31, P41) et 3 cas d'inclusion d'artefacts (P14, P15, P46) qu'un acteur social juge faux. Dans 4 cas (P14, P15, P19, P41) sur 11, une correction probable du problème a été proposée dans la littérature.

TABLE 2 - Problèmes de contradiction sociale

Cas de problème de contradiction sociale	Formalisation
P05 - Wrong inverse relationship	$O \models R^{-1} \equiv R_i, SAP_O \not\models R^{-1} \equiv R_i$
P14 - Misusing "Owl :allvaluesFrom"	$O \models C \sqsubseteq \forall R.C_i, SAP_O \not\models C \sqsubseteq \forall R.C_i, SAP_O \models C \sqsubseteq \exists R.C_j,$
P15 - Misusing "not some" and "some not"	$O \models C \sqsubseteq \neg(\exists R.C_j), SAP_O \not\models C \sqsubseteq \neg \exists R.C_j, SAP_O \models C \sqsubseteq \exists R.\neg C_j$
P19 - Swapping intersection and union	$O \models R.range \equiv (C_i \sqcap C_k), SAP_O \not\models R.range \equiv (C_i \sqcap C_k), SAP_O \models R.range \equiv (C_i \cup C_k)$
P25 - Define a relationship inverse to itself	$O \models R^{-1} \equiv R, SAP_O \not\models R^{-1} \equiv R$
P27 - Defining wrong equivalent relationship	$O \models R_i \equiv R_j, SAP_O \not\models R_i \equiv R_j$
P28 - Defining wrong symmetric relationship	$O \models Symmetric(R), SAP_O \not\models Symmetric(R)$
P29 - Defining wrong transitive relationship	$O \models Transitive(R), SAP_O \not\models Transitive(R)$
P31 - Defining wrong equivalent classes	$O \models C_i \equiv C_j, SAP_O \not\models C_i \equiv C_j$
P41 - DisjointnessOfComplement	$O \models C_i \equiv \neg C_j, SAP_O \not\models C_i \equiv \neg C_j, SAP_O \models C_i \sqcap C_j \equiv \perp$
P46 - Semantic inconsistency	$O \models C_i \sqsubseteq R.C_i, SAP_O \not\models C_i \sqsubseteq R.C_i$

4. Méthode de détection des candidats à l'insatisfiabilité et à la contradiction sociale

En se basant sur le principe qu'un problème d'insatisfiabilité « peut impliquer l'existence de » un problème de contradiction sociale, nous avons extrait, à partir de la formalisation de chacun des 8 cas différents d'insatisfiabilité, les axiomes pouvant représenter chacun une contradiction sociale (i.e. un axiome pouvant être faux selon un acteur social). Par exemple, P36 où $F(P36) = \{C \sqsubseteq \forall R.C_i, C \sqsubseteq \exists R.C_j, C_i \sqcap C_j \equiv \perp\}$, peut impliquer l'existence d'une ou de plusieurs contradictions sociales parmi les suivantes : $(SAP_O \not\models C \sqsubseteq \forall R.C_i)$, $(SAP_O \not\models C \sqsubseteq \exists R.C_j)$ ou $(SAP_O \not\models C_i \sqcap C_j \equiv \perp)$. Neuf axiomes types ont été extraits (on note cet ensemble A_{Ins}). Trois axiomes parmi eux représentent des problèmes de contradiction sociale connus (P14, P15, P31). Nous avons ordonné ces axiomes selon l'ordre décroissant de notre estimation de la probabilité qu'un axiome corresponde à une contradiction sociale. Cette estimation dépend de la difficulté de la compréhension d'un axiome (critère évoqué dans (Roussey et al. 2010)) et de la pertinence de sa considération dans un processus de validation Ce deuxième critère a été jugé en fonction de la rareté d'un axiome dans une ontologie et de son appartenance à plusieurs cas d'insatisfiabilité. L'ordre obtenu (noté $Ord_{A_{Ins}}$) est le suivant : A1 : $C_i \sqcap C_j \equiv \perp$, A2 : $C_i \sqsubseteq \exists R.C_j$, A3 : $C_i \sqsubseteq \forall R.C_j$, A4 : $C_i \sqsubseteq \langle nR.C_j, n > 1$, A5 : $C_i \sqsubseteq \rangle nR.C_j, n > 1$, A6 : $C_i \equiv C_j$, A7 : $R.Domain \equiv C_j$, A8 : $R.Range \equiv C_i$, A9 : $C_i \sqsubseteq C_j$.

A partir des 8 cas différents d'insatisfiabilité, nous avons extrait également 13 anti-patrons partiels de l'insatisfiabilité (APPI). Un APPI est un anti-patron de l'insatisfiabilité auquel on a enlevé un axiome (son axiome manquant). Il aide à identifier une insatisfiabilité et une contradiction sociale : si on est proche d'une insatisfiabilité on est proche d'une contradiction sociale. Un APPI est à utiliser quand les problèmes d'insatisfiabilité sont déjà corrigés dans une ontologie O. En plus, il n'est intéressant que s'il n'y a rien qui contredit son axiome manquant dans O. Dans ce cas, pour engendrer une insatisfiabilité, on complète l'occurrence de chaque APPI (OAPPI) dans O, par l'axiome manquant si la validité de cet axiome est confirmée par un acteur social. Pour détecter une contradiction sociale, on vérifie la validité des axiomes qui composent chaque OAPPI. Dans les deux cas, l'APPI « $C_i \sqsubseteq C_j$ » n'est pas à prendre en compte car il est très fréquent dans une ontologie.

Nous proposons l'heuristique HPCS_APPI (Heuristique pour l'identification des Problèmes de Contradiction Sociale à l'aide des APPI) qui utilise les axiomes de A_{Ins} afin d'aider à cibler des cas de contradiction sociale, tout en cherchant à minimiser l'intervention humaine. Il s'applique à une ontologie ne comprenant pas des problèmes d'insatisfiabilité, sinon il faut tout d'abord les corriger, et il cherche des contradictions sociales dans chaque OAPPI, sans la transformer en une insatisfiabilité.

TABLE 3 - Anti-patrons partiels de l'insatisfiabilité

Anti-patron partiel à 3 axiomes	Axiome manquant
$C_i \cap C_j \equiv \perp, C \sqsubseteq \exists R.C_i, C \sqsubseteq \langle 2R.T$	$C \sqsubseteq \exists R.C_i$
$C_i \cap C_j \equiv \perp, C \sqsubseteq \exists R.C_i, C \sqsubseteq \exists R.C_i$	$C \sqsubseteq \langle 2R.T$
$C \sqsubseteq \exists R.C_i, C \sqsubseteq \exists R.C_i, C \sqsubseteq \langle 2R.T$	$C_i \cap C_j \equiv \perp$
$C_i \cap C_j \equiv \perp, C \sqsubseteq \forall R.C_i,$	$C \sqsubseteq \exists R.C_i$
$C_i \cap C_j \equiv \perp, C \sqsubseteq \exists R.C_i$	$C \sqsubseteq \forall R.C_i$
$C \sqsubseteq \exists R.C_i, C \sqsubseteq \forall R.C_i$	$C_i \cap C_j \equiv \perp$
$C_i \sqsubseteq \exists R.T, R.Domain \equiv C_i$	$C_i \cap C_j \equiv \perp$
$C_i \cap C_j \equiv \perp, R.Domain \equiv C_i$	$C_i \sqsubseteq \exists R.T$
$C_i \cap C_j \equiv \perp, C_i \sqsubseteq \exists R.T$	$R.Domain \equiv C_j$
$C \sqsubseteq \exists R.C_i, R.range \equiv C_i$	$C_i \cap C_j \equiv \perp,$
$C_i \cap C_j \equiv \perp, R.range \equiv C_i$	$C \sqsubseteq \exists R.C_i$
$C_i \cap C_j \equiv \perp, C \sqsubseteq C_i$	$C \sqsubseteq C_j$
$C \sqsubseteq C_i, C \sqsubseteq C_j$	$C_i \cap C_j \equiv \perp,$
$C_i \cap C_j \equiv \perp$	$C_i \equiv C_j$ ou $C_i \sqsubseteq C_j$
$C_i \equiv C_j$	$C_i \cap C_j \equiv \perp$
$C \sqsubseteq \exists R.(C_i \cap C_j)$	$C_i \cap C_j \equiv \perp$
$C_i \sqsubseteq C_j$	$C_i \cap C_j \equiv \perp$

Soit O une ontologie, HPCS_APPI cherche dans O les OAPPI et les classe en trois ensembles F_m (m de 1 à 3), chacun comprend des OAPPI composées du même nombre d'axiomes. Soit n_m le nombre de OAPPI dans F_m . Chaque OAPPI _{m_i} (i de 1 à n_m) correspond à une OAPPI avec m axiomes. Les axiomes A_{ij} de chaque OAPPI _{m_i} (j de 1 à m) sont ordonnés selon l'ordre Ord_{Ains} défini précédemment, tel que si A_{ij} , et A_{ik} deux axiomes de OAPPI _{m_i} et $j < k$ alors A_{ij} appartient à un type d'axiome d'indice inférieur ou égal à celui du type d'axiome auquel appartient A_{ik} . Dans cette heuristique, on traite les OAPPI dans l'ordre décroissant du nombre d'axiomes qui les composent (plus le nombre d'axiomes d'une OAPPI est grand plus la chance de détecter une contradiction dans cette OAPPI est importante) et les axiomes de chaque OAPPI sont vérifiés dans l'ordre Ord_{Ains} . Si un axiome d'une OAPPI est vrai, on indique dans O qu'il est validé et on continue à vérifier les autres axiomes de la même occurrence. S'il est faux, on indique qu'il est faux dans O et on enlève toutes les OAPPI à lesquelles il appartient. HPCS_APP est définie comme suit :

```

Début HPCS_APPI
  Pour  $m= 3$  à 1 FAIRE, Pour  $i=1$  à  $n_m$  Faire
    Vérifier que OAPPI $m_i$  appartient à  $F_m$ , si oui
    Pour  $j=1$  à  $m$ 
      si  $A_{ij}$  est validé alors aller à Fin pour  $j$ 
      sinon vérifier avec un acteur social si  $A_{ij}$  est Vrai
        Si  $A_{ij}$  est faux d'après un acteur social alors
          supprimer  $A_{ij}$  de  $O$  et rajouter ( $\text{not } A_{ij}$ ) à  $O$ 
          si  $A_{ij}$  appartient à un des OAPPI alors  $F=F/OAPPI$ 
          aller à Fin pour  $i$ 
        Sinon noter que  $A_{ij}$  est validé dans tous les OAPPI de  $F$ 
    Fin pour  $j$ 
  Fin pour  $i$ , Fin pour  $m$ 
Fin HPCS_APPI

```

5. Conclusion

Plusieurs travaux se sont intéressés aux problèmes logiques qui nuisent à la qualité d'une ontologie. Les problèmes qui ne peuvent être détectés que par un acteur social sont peu considérés dans la littérature. Dans cet article, nous avons proposé des anti-patrons partiels et

une heuristique pour aider à bien cibler des candidats aux problèmes de contradiction sociale tout en minimisant l'intervention humaine. Actuellement, nous sommes en train d'expérimenter notre approche. Nous cherchons, d'une part, à valider l'intérêt de l'ordre de vérification des axiomes de A_{ins} et celui de traitement des OAPPI pour la minimisation de l'intervention humaine et d'autre part à évaluer le nombre de questions à poser à un acteur social par rapport aux nombres des problèmes corrigés et des problèmes ignorés.

Comme perspectives, nous allons intégrer les APPI et notre heuristique dans notre approche globale de validation d'ontologie (Harzallah et al. 2015), en étudiant à quel moment opportun, il faut identifier et corriger les problèmes de contradiction sociale afin d'améliorer l'efficacité du processus de validation d'ontologie dans sa globalité

Références

- BAUMEISTER J., SEIPEL, D. (2005). Smelly owls-design anomalies in ontologies. pp 215–220 of: Proc. of 18th int. florida artificial intelligence research society conf.
- BUHMANN L., DANIELCZYK, S. & LEHMANN, J. (2011). D3.4.1 report on relevant automatically detectable modelling errors and problems. Tech. rept. LOD2-Creating Knowledge out of Interlinked Data.
- FAHAD M., M.ABDUL QADIR & S. A. H. SHAH (2008). Evaluation of ontologies and dl reasoners. In Zhongzhi Shi, E. Mercier-Laurent, and D. Leake, editors, Intelligent Information Processing IV, volume 288 of IFIP : The International Federation for Information Processing, pp 17–27.
- GHÉRASIM T.D. (2013). Détection de problèmes de qualité dans des ontologies construites automatiquement à partir de textes. Thèse en Informatique de l'université de Nantes.
- GOMEZ-PEREZ A., FERNANDEZ-LOPEZ, M., CORCHO, O. (2001). Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web. Adv. Inf. And Know. Processing. Springer.
- GUARINO N., OBERLE, D., & STAAB, S. (2009). What is an ontology? pp 1–17 of: Studer, R., & Staab, S. (eds), Handbook on ontologies, 2 edn. International Handbooks on Information Systems. Springer.
- HARZALLAH M., G. BERIO, & KUNTZ P. (2015). Towards an Approach for Configuring Ontology Validation. In Knowledge Discovery, Knowledge Engineering and Knowledge Management Vo.553 of the series Communications in Computer and Information Science. pp 388-404, Springer.
- KROGSTIE, J., O. LINDLAND & G. SINDRE (1995). Defining quality aspects for conceptual models. In Proceedings of the IFIP8.1 Working Conference on Information Systems Concepts : Towards a Consolidation of Views (ISCO3), pp 216–231.
- NEUTHAUS F. & A VIZEDOM. (2013). Towards Ontology Evaluation across the Life Cycle. In ontolog.cim3.net/file/work/OntologySummit2013/OntologySummit2013,_Communiqué.
- PAMMER V. (2010). Automatic Support for Ontology Evaluation Review of Entailed Statements and Assertional Effects for OWL Ontologies. PhD thesis, Graz University of Technology.
- POVEDA M., SUAREZ-FIGUEROA, M.C. & GOMEZ-PEREZ, A. (2012). Validating ontologies with oops !, dans Proceedings of the 18th international conference on Knowledge Engineering and Knowledge Management (EKAW'12), Springer-Verlag, pp. 267– 281.
- QAZI N. I. & M. A. QADIR (2011). Algorithms for the evaluation of ontologies for extended error taxonomy and their application on large ontologies. J. UCS, 17(7) pp:1005–1020.
- RODLER P., K. SHCHEKOTYKHIN, P. FLEISS & FRIEDRICH G. (2013). Rio: Minimizing user interaction in ontology debugging. In Wolfgang Faber and Domenico Lembo, editors, Web Reasoning and Rule Systems, volume 7994 of Lecture Notes in Computer Science, pp 153–167. Springer Berlin Heidelberg.
- ROUSSEY C., F. SCHARFFE, O. CORCHO, AND O. ZAMAZAL. (2010). Une méthode de débogage d'ontologies OWL basées sur la détection d'anti-patterns. In Sylvie Desprès, editor, IC2010, 21ème Journées francophones d'Ingénierie des Connaissances, Collection mathématiques & Informatique, pp 43–54. Presses de Nimes.
- WANG H., M. HORRIDGE, A. RECTOR, N. DRUMMOND & J SEIDENBERG (2005). Debugging owl-dl ontologies: A heuristic approach. In Y Gil, E Motta, V.R Benjamins, and M.A. Musen, editors, The Semantic Web, ISWC 2005, volume 3729 of Lecture Notes in Computer Science, pp 745–757. Springer Berlin Heidelberg.