

ElvirIA-P : génération d'avis d'expertise pour accompagner les experts en sûreté de fonctionnement des logiciels critiques

K. Cousot¹, T. Sanchez¹, A. Nguyen¹, A. Calpas², G. Martinez², C. Lopez¹

¹ Emvista, Cap Oméga, Rond-point Benjamin Franklin, 34960 Montpellier

² Agence Innovation Défense (AID), 60 boulevard du Général Martial Valin - CS 21623 - 75509 Paris cedex 15

DGA Techniques Aéronautiques, 47 Rue Saint-Jean, 31130 Balma

{prenom.nom}@emvista.com ; {prenom.nom}@intradef.gouv.fr

Résumé

ElvirIA-P est une preuve de concept ayant pour objectif d'aider l'expert tout au long du processus de certification pour les logiciels embarqués critiques pour la sûreté aéronautique. La solution consiste à extraire des informations d'intérêt pour l'expert dans la masse de documents à sa disposition à partir desquelles un avis d'expertise est généré. Les travaux ont d'abord consisté à représenter les connaissances nécessaires aux experts pour générer un avis d'expertise, puis à extraire les informations des textes pour peupler la base de connaissance et appliquer un raisonnement qui permet de déduire des nouvelles connaissances.

Mots-clés

Preuve de concept, ontologie, raisonnement.

Abstract

ElvirIA-P is a proof of concept aiming at helping the expert throughout the certification process for critical on-board software for aeronautical security. The solution consists in extracting information of interest for the expert in the whole technical documents from which an expert opinion is generated. The work consisted in representing the knowledge necessary for the experts to generate an expert opinion, then in extracting information from the texts to populate the knowledge base and apply reasoning to deduce new knowledge.

Keywords

Proof of concept, ontology, reasoning.

1 Introduction

Au sein de la Direction Générale pour l'Armement (DGA), les experts en sûreté de fonctionnement, du site de DGA Techniques Aéronautiques (DGA TA), spécialisés en logiciel critique doivent faire face à des contraintes fortes : temps limité pour évaluer les logiciels, pénibilité de certaines tâches telles que la lecture de documents volumineux qui nécessite une connaissance *a priori* de formalismes et de langues différents, champs d'expertises vastes, etc.

ElvirIA-P est une preuve de concept, financée et encadrée par l'Agence Innovation Défense (AID), ayant pour objectif d'aider l'expert tout au long du processus de certification des logiciels. La solution consiste à extraire des informations d'intérêt pour l'expert dans la masse de documents à sa disposition à partir desquelles un avis d'expertise est généré. Pour l'outil ElvirIA-P, il s'agit d'évaluer les moyens de conformité décrits dans le corpus documentaire pour un logiciel donné. Précisons que les utilisateurs d'ElvirIA-P sont les autorités de certification et non les concepteurs de système. Seul un sous-ensemble du corpus documentaire est transmis à l'autorité et c'est à partir de ce sous-ensemble que l'autorité établit une première évaluation des moyens de conformité proposés par l'appliquant. Les audits et les inspections du corpus documentaire chez l'appliquant vont lui permettre d'établir son avis final. Précisons encore que les auditeurs logiciel (autorité de certification) sont totalement indépendants des développeurs du logiciel. ElvirIA est un outil d'aide pour l'autorité de certification pour l'avis d'expertise.

Dans notre contexte métier qui s'inscrit dans l'aéronautique, le corpus documentaire est constitué des PSAC (*Plan for Software Aspects of Certification*) et SDP (*Software Development Plan*) qui sont les documents décrivant un projet (par exemple le développement d'un logiciel embarqué sur un moteur d'hélicoptère) rédigés par des industriels tels que Thales ou Airbus. Ces documents sont soumis aux experts de la DGA TA qui évaluent comment l'appliquant a pris en compte les recommandations du référentiel métier aéronautique, DO-178, qui énonce un ensemble d'objectifs sur les méthodes et les outils pour le développement d'un logiciel embarqué. Les experts effectuent la mesure de cette conformité par des évaluations des données du cycle de vie du logiciel. L'évaluation se fait généralement en 4 grands audits (planification, développement, vérification et finale). Les documents de certification et les audits permettent alors un audit logiciel et vérifient que les objectifs définis dans le DO-178 sont atteints avant de délivrer un certificat.

Ainsi, pour ElvirIA-P, les textes à analyser sont contenus dans les PSAC et SDP et doivent être confrontés au référentiel métier DO-178. Le développement de ElvirIA-P nécessite :

- une ontologie qui représente les connaissances des experts et les règles métiers associées ;
- un moteur d'extraction d'information faisant appel à des techniques du Traitement Automatique du Langage Naturel pour peupler la base de connaissance ;
- un raisonneur capable d'inférer des connaissances non explicites dans le corpus documentaire ;
- un générateur d'avis d'expertise à partir des données extraites et inférées ;

4. Le DS prétraité est transmis à un analyseur syntaxique et sémantique développé par Emvista pour analyser et extraire les informations pertinentes. L'utilisateur peut voir les éléments identifiés dans les documents via l'interface Web et les corriger si nécessaire avant de les envoyer au moteur de raisonnement ;
5. Une fois les éléments validés par l'utilisateur, le moteur de raisonnement est lancé ;

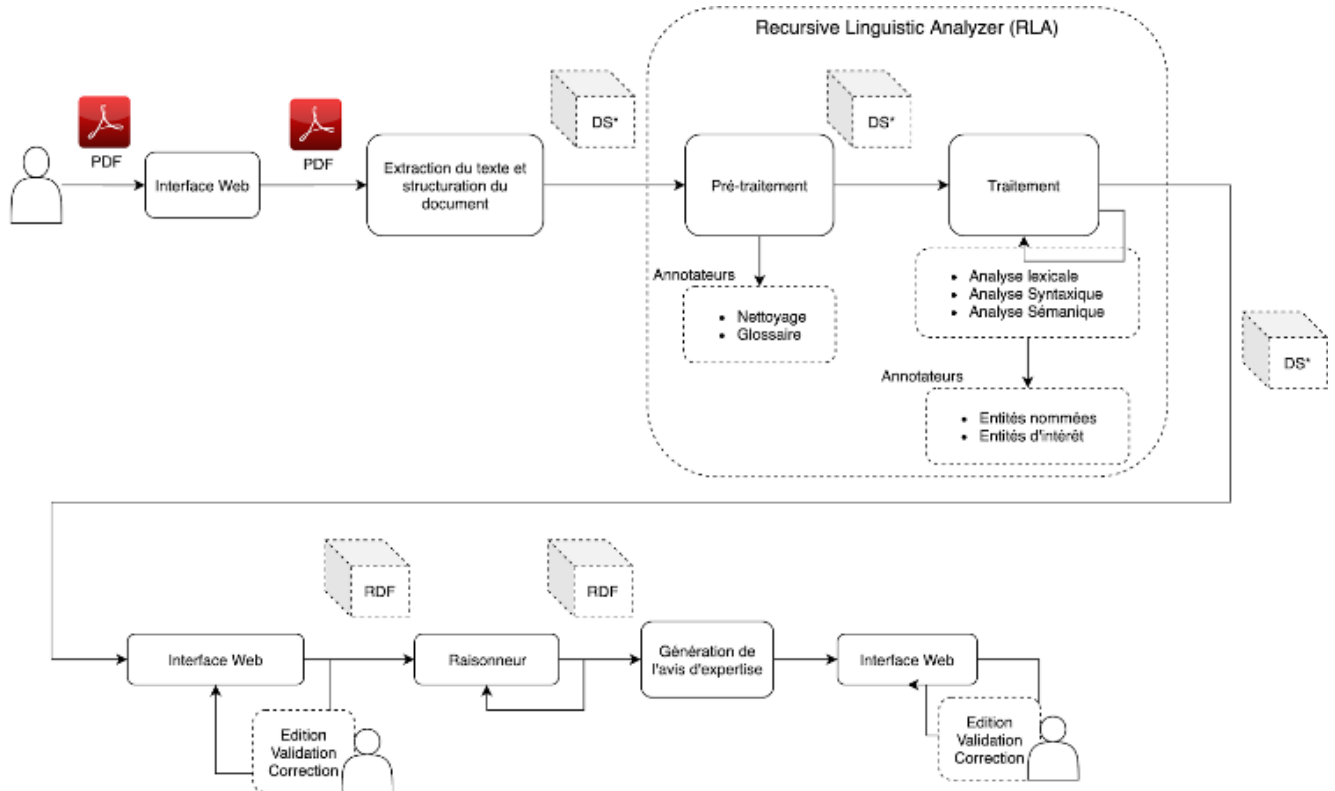


Fig. 1 : Schéma du workflow applicatif

Cet article se focalise sur l'architecture de la solution (cf. section 2), la conception et le développement de l'ontologie.

2 Architecture et flux opérationnel

La conception de l'architecture de ElvirIA-P a été guidée par le flux opérationnel répondant au besoin exprimé par DGA. Le workflow applicatif est décrit de façon séquentielle comme suit (cf. Fig. 1) :

1. L'utilisateur crée un projet sur l'application Web et fournit en entrée un ou plusieurs PSAC/SDP au format PDF ;
2. L'application extrait le contenu textuel et le segmente (sections, paragraphes, ...). L'application retourne un Document Structuré, noté DS dans la suite ;
3. Le DS est soumis à un premier traitement de nettoyage (suppression des données trop bruitées, des lignes vides ou avec trop peu de caractères). Cette étape fournit le DS prétraité ;

6. L'avis d'expertise est généré ;
7. L'utilisateur peut encore une fois éditer ou compléter les données générées avant d'enregistrer l'avis.

Compte tenu de ce scénario, une interface Web a été développée avec pour objectif de :

1. permettre à l'utilisateur de modifier des données extraites automatiquement par l'analyseur sémantique avant d'envoyer ces données au raisonneur ;
2. permettre à l'utilisateur de visualiser et d'éditer les avis d'expertises générés automatiquement ;

3 Conception et développement de l'ontologie

Un des premiers objectifs était de développer l'ontologie métier représentant toutes les classes utiles au raisonnement de l'expert

pour générer un avis d'expertise pertinent. Nous avons adopté la méthodologie de Ushold et Gruninger (1996) consistant à identifier les scénarios et les questions de compétences, c'est-à-dire les questions auxquelles notre ontologie devait être en mesure de répondre pour fournir du contenu à l'avis d'expertise. Les scénarios et les questions de compétences ont été élaborés en étroite collaboration entre les ingénieurs de recherche de Emvista et les experts DGA.

Un exemple de scénario est l'identification par un expert des activités traitées/non traitées dans les PSAC/SDP, ce qui de fait bloque la délivrance du certificat. Un autre exemple consiste à repérer les techniques mises en œuvre dans les PSAC/SDP et voir si chacune d'entre elles traite l'ensemble des méthodes nécessaires ainsi que les activités relatives à ces méthodes.

Dans le but de déterminer les spécifications de l'ontologie, des questions de compétences ont été identifiées (Gruninger et Fox, 1995), par exemple : « Quelles activités ont été traitées dans les PSAC/SDP ? » ; « Dans quelle section est décrite la méthode A.7-2 ? » ; « Est-ce que la technique B est partiellement traitée ? ».

Ces questions ont permis d'identifier onze *classes* définies sur deux niveaux, six *object properties* et onze *data properties*.

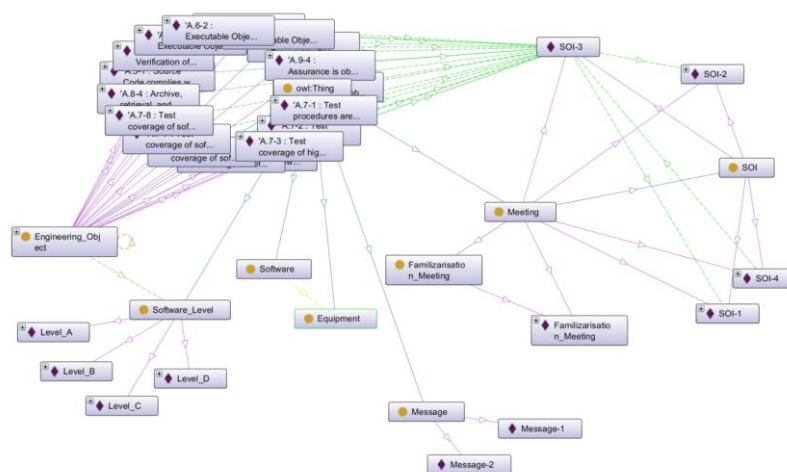


Fig. 2 : Aperçu de l'ontologie développée.

Trois classes essentielles sont apparues immédiatement et ont été définies comme suit :

1) *Technique* : une Technique est un ensemble de Méthodes associé à un savoir-faire métier. Les Techniques sont mises en œuvre par l'équipe d'ingénierie développant le système embarquant un/des produits logiciel dans le but d'atteindre la certification. Le nombre de techniques est connu, il en existe huit nommées de A à H. Exemples de techniques : A, « Technique d'aspect système liés au développement logiciel » ; B, « Technique d'organisation des processus du cycle de vie

du produit logiciel »)

2) *Méthode* : Une Méthode est un ensemble d'activités qui doivent être correctement traitées afin d'être validée. Le nombre de méthodes est fixé à 74. Exemples de méthodes : A.7-2, « Les résultats des tests sont corrects et les écarts expliqués. » ; A.4-10, « L'architecture logicielle est compatible avec l'ordinateur cible ».

3) *Activité* : Une activité est un critère d'éligibilité qui agrège des conditions à respecter lors du développement d'un logiciel. Exemple d'activité : Act-5.1.2.a « Les exigences fonctionnelles du système et les exigences d'interface système qui sont allouées au logiciel ne sont pas ambiguës, incohérentes et ne contiennent pas de conditions indéfinies.

D'autres classes permettent de représenter les types d'auditions organisées entre les industriels et les experts DGA (classe Meeting et ses sous-classes, par exemple Familiarization Meeting et Stages_of_Involvement) ou encore les niveaux de criticité des logiciels (*Software_Level*).

Les *object properties* identifiées permettent de lier les entités entre elles, avec notamment des relations de méronymie (entre

Software et *Equipment* par exemple) ou des relations transitives (*is_dependent_of*, entre une Méthode et une Technique et entre une Activité et une Méthode), symétriques (*is_related_to*, entre deux *Engineering_Object*) ou sans propriétés spécifiques (*has_software_level*, entre *Engineering_Object* et *Software_Level*).

Étant donné que les techniques, les méthodes et les activités sont prédéfinies et n'ont pas vocation à évoluer, elles sont considérées comme des individus nommés : plus de 600 individus sont présents dans la base. Une vue d'ensemble des concepts, relations et individus est donnée en Figure 2. Le développement de l'ontologie a été réalisé avec le logiciel Protégé¹.

¹ <https://protege.stanford.edu/>

4 Acquisition et développement des règles métiers

Les règles métiers utilisent les concepts de l'ontologie pour générer des informations qui seront restituées dans l'avis d'expertise. Ces règles ont été acquises par le biais de documents textuels de type « référentiels métiers » (précisément la DO-178C qui fixe les conditions de sécurité applicables aux logiciels critiques de l'avionique dans l'aviation commerciale et l'aviation générale) et directement par le biais des experts.

La compréhension du besoin des experts s'est réalisée au cours de nombreux échanges afin de représenter de façon formelle le raisonnement appliqué pour émettre un avis d'expertise. Il en est ressorti que deux familles de règles devaient être prises en compte :

- les règles notées EXP : il s'agit des règles issues du savoir-faire des experts, de leur expérience. Ces règles ne sont explicitées dans aucun document textuel et ont donc été formalisées au cours des discussions.
- les règles REF : il s'agit de règles explicitement décrites dans les référentiels des experts. La difficulté de ces règles est qu'elles sont exprimées en langage naturel et sont difficilement interprétables, y compris par les experts à cause de leur caractère subjectif. Exemple de règle issue de la DO-178C : « *IF a different compiler or different set of compiler options are used, resulting in different object code, THEN the results from a previous software verification process activity using the object code may not be valid and must not be used for the new application. In this case, previous test results may no longer be valid for the structural coverage criteria of the new application. Similarly, compiler assumptions about optimization may not be valid. Therefore, some software verification process activities have to be done again.* ».

Étant donné la complexité des règles REF, chacune d'entre elles est représentée par un individu nommé et est liée à l'activité correspondante de sorte à demander à l'expert d'appliquer cette règle le cas échéant via l'avis d'expertise. À noter que les identifiants des paragraphes et de pages de la DO_178C sont également associés à chaque règle, et font partie intégrante de l'ontologie. Leur traitement automatique demeure une perspective à ce travail.

D'une part, les règles REF ont été intégrées dans l'ontologie sous forme textuelle (cf. Fig. 1) et non sous forme de règles prises en compte dans le raisonnement pour les raisons évoquées dans la section précédente. Néanmoins, les relations d'association transitives entre la Règle et les Techniques et Méthodes sont prises en compte dans le raisonnement.

D'autre part, les règles EXP ont été développées dans l'ontologie en logique de descriptions avec le langage de programmation SWRL. Ces règles sont donc prises en compte dans le raisonnement d'ElvirIA-P, contrairement aux règles

REF. Exemples de règles EXP : Si une activité n'est pas traitée alors la méthode liée à cette activité n'est pas validée (cf. (1) Fig. 4) ; si deux logiciels de niveau de criticité différents sont détectés alors le message 1 doit être retourné (cf. (2) Fig. 4).

```
1) ontology:needs(?m, ?a) ^ ontology:Activity(?a) ^
ontology:is_validated(?a, false) ^ ontology:Method(?m) ->
ontology:is_validated(?m, false)

2) ontology:Software(?s) ^ ontology:has_software_level(?s,
?dal1) ^ ontology:has_software_level(?s, ?dal2) ^
differentFrom(?dal1, ?dal2) ->
ontology:is_validated(autogen1:Message-1, true)
```

Fig. 4 : Exemples de règles métiers développées en SWRL

5 Acquisition des données et prétraitement

Après un traitement consistant à sélectionner les documents PSAC/SDP en anglais, au format PDF et non scannés, le corpus d'étude contient 73 fichiers.

La solution Apache PDFBox a été adoptée pour extraire le texte des PDF ; celle-ci prend en entrée un fichier PDF et fournit le texte brut correspondant ainsi que le texte formaté avec HTML. La sortie HTML n'a pas été retenue dans la suite du traitement car celle-ci n'est pas assez fiable (problème fréquent avec les balises <p>). Un module a été développé pour fournir la structure qui contient le texte brut ainsi que son découpage en pages. Celui-ci est composé du Segmenter et du Cleaner décrit ci-après.

Segmenter. Le Segmenter a pour objectif d'identifier des segments textuels particuliers au sein du document. Certains segments seront supprimés par la suite afin de limiter le bruit tandis que d'autres pourront être exploités. C'est d'abord la table des matières qui est recherchée à l'aide de patrons, puis, grâce à l'information qu'elle contient, la structure du document en sections est reconstituée.

Le Segmenter détecte ensuite les en-têtes et les pieds de page : un score de similarité est calculé avec les pages avoisinantes pour mettre en évidence les éléments fortement redondants en début et fin de page. Chacun des segments est ensuite envoyé au Cleaner.

Cleaner. L'objectif du Cleaner est de supprimer le bruit dans le texte extrait. Le bruit trouve son origine dans des défauts du traitement OCR ainsi que dans l'extraction du texte des tableaux et des figures qui ne se présente pas sous la forme d'une phrase analysable par des outils classiques de TALN. Deux méthodes ont été envisagées, l'une faisant partie de la famille d'approche "symbolique" et l'autre faisant partie de la famille d'approche "par apprentissage". La première s'est avérée tout à fait pertinente ce qui a impliqué que la deuxième option a été abandonnée, d'autant plus que celle-ci aurait

nécessité un grand volume de données annotées que nous n'avions pas.

Une fois que le texte est segmenté et nettoyé, il est envoyé au module d'extraction d'information pour peupler la base de connaissances.

6 Peuplement de la base de connaissance

Étant donné l'ontologie développée, le peuplement de la base de connaissance s'opère *via* deux tâches :

- L'extraction des entités nommées qui permettent de définir le contexte de l'avis d'expertise (numéro de version du document, nom des logiciels concernés, *etc.*) ;
- L'identification des activités traitées à partir desquelles des inférences s'opèrent et qui constituent le fond de l'avis d'expertise ;

La reconnaissance d'entités nommées consiste en un token-level classifier (Devlin et al., 2018) avec une couche de classification après les encodeurs des modèles transformers (Vaswani et al., 2017). Cette couche est une couche dense qui prend en entrée une représentation du "First Sub-Token" (Devlin et al., 2018) et qui fournit une classe (personne, localisation ou organisation) pour chaque mot donné au modèle.

Une surcouche symbolique permet d'identifier les entités nommées pour lesquelles trop peu de données sont annotées (par exemple les noms de logiciels et des équipements).

L'identification des activités est effectuée par la détection de descripteurs spécifiques à une activité (fournis par DGA) qui prennent la forme de mots clés et de triplets exprimés en langage naturel avec utilisation d'opérateurs logiques AND et OR. Concernant les mots-clés, une correspondance est recherchée entre les mots-clés fournis et les mots dans le texte qui sont soumis à une phase de lemmatisation afin de considérer

Proposition d'avis d'expertise du 18/06/2021 10:50

L'analyse des documents suivants par ElvirIA-P fait état de plusieurs risques pour le projet **Projet1**.

- ATR NAS PSAC IADSW J55740AF 00 (PSAC)
- 332A884823 A PSAC AMC plus V3 (PSAC)

Projet non confidentiel

Message(s) d'avertissement :

⚠ S'assurer que les activités mentionnées dans le plan (SDP/PSAC) sont bien rattachées à leur logiciel ou à leur niveau de DAL respectif affiché dans le plan.

Liste SOI :

- SOI-1 (non validé)
- SOI-2 (non validé)
- SOI-3 (non validé)
- SOI-4 (non validé)

Liste des équipements :

MCDU
AFDX

Liste des techniques :

Technique_A	Non couverte
Technique_B	Non couverte
Technique_C	Partiellement couverte
Technique_D	Non couverte
Technique_E	Partiellement couverte
Technique_F	Partiellement couverte
Technique_G	Partiellement couverte
Tech-G : Technique de gestion de configuration des données d'ingénierie relative au produit logiciel	
Méthode A-8-1	Couverte
Méthode A-8-2	Partiellement couverte
Méthode A-8-3	Partiellement couverte
Méthode A-8-4	Partiellement couverte
Méthode A-8-5	Couverte
Méthode A-8-6	Partiellement couverte
Technique_H	Partiellement couverte

Fig. 4 : Exemple d'une page d'une proposition d'avis d'expertise générée par ElvirIA-P

```
<owl:NamedIndividual rdf:about="https://www.defense.gouv.fr/dga/elviriap/ontology/activity#Act-5.3.2.b">
  <rdf:type rdf:resource="https://www.defense.gouv.fr/dga/elviriap/ontology/Activity"/>
  <rdf:type rdf:resource="https://www.defense.gouv.fr/dga/elviriap/ontology/Engineering_Object"/>
  <is_validated rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</is_validated>
  <rdfs:comment xml:lang="en">The &quot;Source Code&quot;, conforms to the &quot;Software Code Standards&quot;.</rdfs:comment>
</owl:NamedIndividual>
```

Fig. 3 : Exemple de triplets RDF relatifs à une activité détectée dans un segment textuel.

La méthode symbolique s'appuie sur des hypothèses de la langue au niveau lexical ; par exemple il est fortement probable qu'une phrase contienne un déterminant ou une majuscule en début de phrase ou encore un point en fin de phrase. Au contraire, la présence soutenue de caractères spéciaux dans une chaîne de caractères tend à montrer que celle-ci n'est pas une phrase bien formée et peut-être le résidu d'un tableau ou d'un schéma, par exemple. Des travaux spécifiques à l'exploitation des figures et des tableaux devraient être menés, par exemple en utilisant Dagobah (Liu et Troncy, 2019). Dans ce module, il n'est pas envisageable d'utiliser une analyse plus profonde du texte (syntaxe, sémantique) puisqu'il s'agit ici justement de préparer le texte à une telle analyse.

les formes fléchies. Exemple de mots-clés pour l'activité Meth-2.1 :

"software components" OR "several software components" / "components" OR "several components" OR "partitioning" / "partition" OR "software partition"

Concernant les relations, une analyse syntaxique du texte est réalisée afin d'identifier les verbes pour lesquels une correspondance est recherchée avec le prédicat du triplet fourni. Une fois le prédicat identifié, son sujet et son objet sont recherchés dans la phrase via les relations syntaxiques « sujet » et « objet ». Cette méthode permet d'identifier des triplets dont

les éléments sont distants dans les textes. Exemple de triplets pour l'activité Act-5.3.2.a :

"source code implement low-level requirement" AND "source code comply with software architecture" OR "source code implement requirement" AND "source code comply with software architecture"

Afin d'augmenter la couverture du traitement, la détection de descripteurs spécifiques s'appuie également sur un niveau sémantique via l'utilisation de synonymes pour les sujets et les objets et de classes sémantiques de VerbNet pour les verbes (Schuler, 2005).

Lorsqu'une information relative à une activité est détectée, ladite activité reçoit la valeur *true*. Un exemple de RDF représentant une activité détectée est donné en Fig. 3. Les activités traitées déclenchent des règles en cascade jusqu'à déterminer si les techniques sont traitées et si les auditions peuvent être organisés.

Nous n'aborderons pas ici la génération de l'avis d'expertise à proprement dite mais présentons un exemple (cf. Fig. 4) généré à partir des données structurées et présentes dans la base de connaissance. En quelques clics, l'expert peut accéder à la source (ou aux sources) ayant généré chaque élément présent dans l'avis. Il accède également aux règles EXP impliquées dans ladite génération. Dans le concept ElvirIA, l'objectif est d'évaluer comment une IA peut aider l'auditeur à construire son avis d'expertise.

7 Conclusion

Dans cet article, nous avons présenté ElvirIA-P, une preuve de concept ayant pour objectif d'aider l'expert tout au long du processus de certification pour les logiciels embarqués critiques pour la sûreté aéronautique. Les travaux ont d'abord consisté à représenter les connaissances nécessaires aux experts pour capitaliser, structurer et générer un avis d'expertise, puis à extraire les informations des textes pour peupler la base de connaissances et appliquer un raisonnement qui permet de déduire des nouvelles connaissances.

Des tests du raisonneur ont été réalisés avec différents algorithmes de raisonnement tels que Pellet et Fact++. Ils démontrent que l'ontologie est consistante ; aucun problème d'incompatibilité des règles n'a été détecté.

Les requêtes SPARQL mises en œuvre pour répondre au besoin de la génération d'avis pour les experts de la DGA ont toutes retournées les résultats attendus et permettent effectivement de faire remonter les informations jusqu'à l'interface utilisateur. Les inférences obtenues sont bien celles attendues. Celles-ci sont réalisées en quelques millisecondes.

La preuve de concept n'est cependant pas industrialisable en l'état. La preuve de concept nécessite des études et des travaux de développement supplémentaires pour pouvoir raisonnablement envisager de la mettre au service d'un expert pour l'accompagner dans son travail. Le projet ElvirIA a montré

que les techniques IA peuvent assister dans une certaine mesure le travail de l'expert. Les échanges entre les auditeurs DO-178 et Emvista ont permis de mieux appréhender les possibilités et les limitations des technologies IA disponibles sur le marché sur le traitement du langage. Ils ont mis en évidence que la modélisation de notre métier sur les caractéristiques des technologies du logiciel était très complexe pour l'apprentissage d'une machine contrairement à un référentiel de développement logiciel.

L'analyseur utilisé étant un outil d'analyse de texte bien formée, il a été décidé de se focaliser prioritairement sur les phrases nominales et verbales. Les images, tableaux et listes contiennent néanmoins des informations qui pourront s'avérer indispensables pour la génération d'avis d'expertise et pourront faire l'objet d'un traitement prioritaire dans les prochains travaux, par exemple en utilisant un système d'annotation sémantique de données tabulaires tel que DAGOBAN [7].

8 Références

- [1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- [2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.
- [3] M. Gruninger et M. S. Fox, *Methodology for the Design and Evaluation of Ontologies*, 1995.
- [4] Liu, J., & Troncy, R. (2019). DAGOBAN: An End-to-End Context-Free Tabular Data Semantic Annotation System.
- [5] Schuler, K. K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon.
- [6] M. Uschold et M. Gruninger, *Ontologies: Principles, methods and applications. The knowledge engineering review*, 11(02), pp. 93-136, 1996.
- [7] Chabot, Y., Labbé, T., Liu, J., & Troncy, R. (2020). DAGOBAN: Un système d'annotation sémantique de données tabulaires indépendant du contexte. In *31es Journées franco-phones d'Ingénierie des Connaissances (Proceedings of the 31st French Knowledge Engineering Conference)*, Angers, France, June 29 - July 3, 2020, pp. 120-132.