

Architecture d'un Système Distribué pour l'Annotation Assistée de Corpus Vidéo

Christophe COLLET, Matilde GONZALEZ, Fabien MILACHON

IRIT (UPS - CNRS UMR 5505)

Université Paul Sabatier, 118 Route de Narbonne, F-31062 TOULOUSE CEDEX 9

{collet, gonzalez, milachon}@irit.fr

Résumé. Cet article présente un composant du projet Dicta-Sign - projet européen FP7 ICT d'une durée de trois ans - qui vise à améliorer la communication web dans la communauté sourde. Une part de ce projet concerne l'annotation de corpus vidéo de la langue des signes. Afin d'améliorer les tâches d'annotation en termes de temps et de reproductibilité, plusieurs *plugins* pour le traitement des vidéos de la langue des signes ont été développés. Le composant présenté dans ce document lie différents *plugins* aux logiciels d'annotation au travers du réseau. Ces *plugins* peuvent être développés dans différents langages de programmation, systèmes d'exploitation et ordinateurs. Pour ce faire, le *Webservice* SOAP ainsi qu'un format spécifique des données en XML pour l'échange sont utilisés.

Abstract. This paper present one component of Dicta-Sign, a three-year FP7 ICT project that aims to improve the state of web-based communication for Deaf people. A part of this project is the annotation of sign language corpora. To improve the annotation task in terms of reproducibility and time consuming, several plug-ins for sign language video processing are developed. The component presented in this paper aims to link several plug-ins to annotation software through the network. These plug-ins can be coded in different languages, operating systems and computers. For that, it uses the SOAP Webservice and a specific data-format in XML for the data exchange.

Mots-clés : outil d'annotation, traitement automatique des vidéos, langue des signes, système distribué.

Keywords: annotation tool, automatic video processing, sign-language, distributed system.

1 Introduction

Actuellement, de nombreuses recherches traitent de l'analyse et de la reconnaissance de la langue des signes, avec pour but de comprendre, reproduire ou traduire celle-ci (Ong et Ranganath, 2005). Dans le domaine de l'informatique, ces recherches portent sur le développement de traitements automatiques de vidéo en langue des signes (Lefebvre-Albaret et Dalle, 2009; Theodorakis *et al.*, 2009). L'évaluation de ces traitements requiert des corpus vidéos représentant plusieurs heures de langue des signes. Ces vidéos sont généralement annotées manuellement par des linguistes et des chercheurs en informatique. Différents outils d'annotation (*Annotation Tool* : AT) ont été développés pour réaliser cette tâche, comme par exemple Elan (Wittenburg *et al.*, 2006), Anvil (Kipp, 2001), Ilex (Hanke, 2002; Hanke et Storz, 2008) et

AnColin (Braffort *et al.*, 2004). Dans le cas de vidéos de longue durée, l'annotation manuelle est source d'erreurs, non reproductible et chronophage. De plus, la qualité des annotations dépend grandement de l'expertise de l'utilisateur. L'association de cette expertise à des traitements automatiques facilite cette tâche et représente un gain de temps. C'est pourquoi nous proposons un système d'intégration de traitements automatiques (Assistant Automatique d'Annotation : A^3), aux ATs existants.

Du point de vue de l'utilisateur, l'exploitation d'outils d'aide à l'annotation (comme un A^3) doit être simple et rapide. L'utilisateur doit pouvoir extraire une partie de la vidéo et utiliser une annotation préalablement définie comme paramètre en entrée d'un A^3 . La figure 1 schématise l'interface d'un AT. L'utilisateur a réalisé des annotations sur les deux pistes AG1 et AG2 (fig. 1a). Lorsqu'il invoque un traitement automatique, nécessitant par exemple deux paramètres d'entrée (fig. 1b.), les deux nouvelles pistes à remplir apparaissent : P1 et P2. Celles-ci peuvent être remplies manuellement ou par copie d'autres pistes. Enfin, lorsque le traitement est terminé, P1 et P2 disparaissent pour laisser place aux pistes représentant les résultats du traitement ; R1 (fig. 1c). De même manière, cette piste peut être copiée sur une nouvelle piste AG3 par exemple, pour la sauvegarder et/ou la modifier.

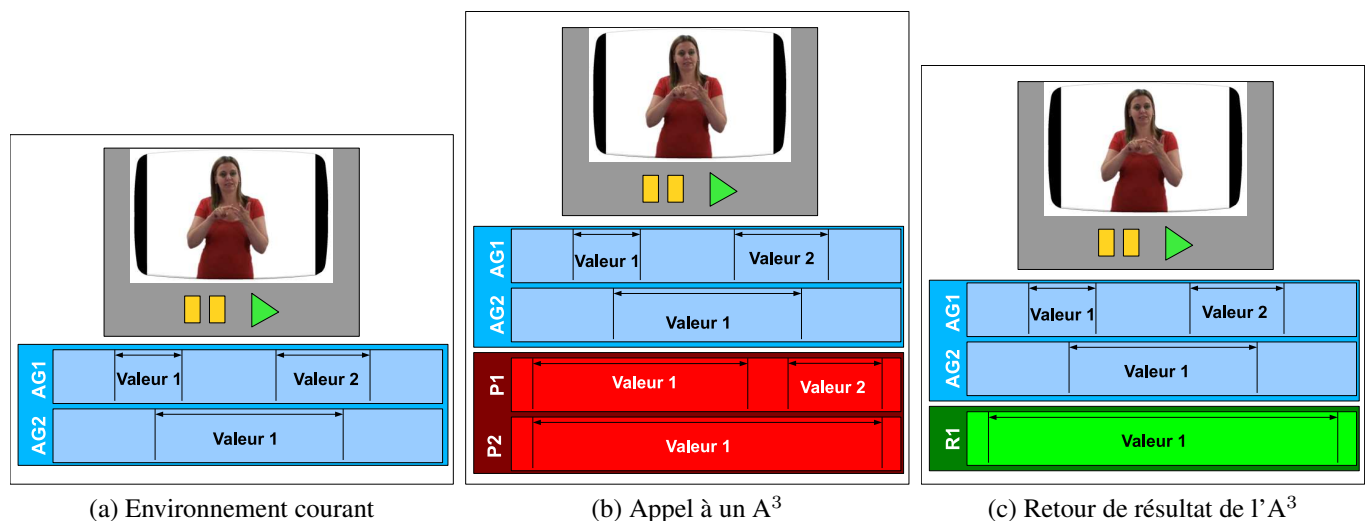


FIGURE 1: Exemple d'utilisation d'un A^3 dans un AT.

La difficulté d'intégration des assistants automatiques à l'AT ne réside pas uniquement dans la programmation d'une interface ergonomique, mais aussi dans la communication inter A^3 s-AT. En effet, ces différents programmes sont développés dans des environnements souvent incompatibles. C'est pourquoi nous proposons d'utiliser une architecture distribuée en spécifiant les protocoles de communication et le format des données échangées. Nous décrivons d'abord son architecture et son fonctionnement, puis nous définissons le format de communication des données utilisé. Enfin, nous présentons la bibliothèque de développement utilisée pour ces communications ainsi que les systèmes de sécurité associés.

2 Présentation de l'architecture du système

Le principal problème dans la mise en œuvre d'interactions entre des A^3 s et des ATs est l'incompatibilité des langages de programmation, des systèmes d'exploitation et des architectures matérielles. Cependant,

il est impossible de restreindre les conditions de développement à un environnement unique car les traitements peuvent être très complexes et il est préférable de les développer dans un langage spécifique voire de les exécuter sur des ordinateurs adaptés. C'est pourquoi nous proposons de résoudre ce problème à l'aide d'un système distribué où les A³s peuvent être hébergés sur différents serveurs distants.

La communication et les échanges de données sont donc effectués au travers du réseau à l'aide d'un protocole et d'un format compréhensible par tous les composants du système. Le format des données doit donc être standardisé. Pour cela, les Interfaces de Programmation de l'Application (API) des A³s doivent contenir des paramètres dont la structure est compatible avec la structure des données des ATs. La description des données est réalisée au format XML et structurée sous la forme d'un graphe d'annotation (*Annotation Graph : AG*) (Bird et Liberman, 2001; Schmidt *et al.*, 2008). Ce graphe a une structure similaire à celle utilisée dans les ATs, soit une hiérarchie des pistes, accompagnée d'une liste de valeurs associées à chaque image de la séquence. D'autre part, les ATs sont généralement capables d'importer/exporter des graphes d'annotation. Ces graphes sont représentés dans un format XML, et sont étendus par l'ajout d'informations concernant le processus et le fichier vidéo à traiter.

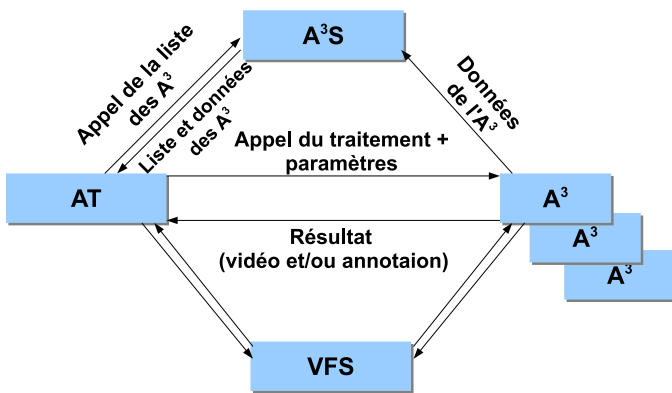


FIGURE 2: Architecture du système distribué pour l'annotation assistée de corpus vidéo

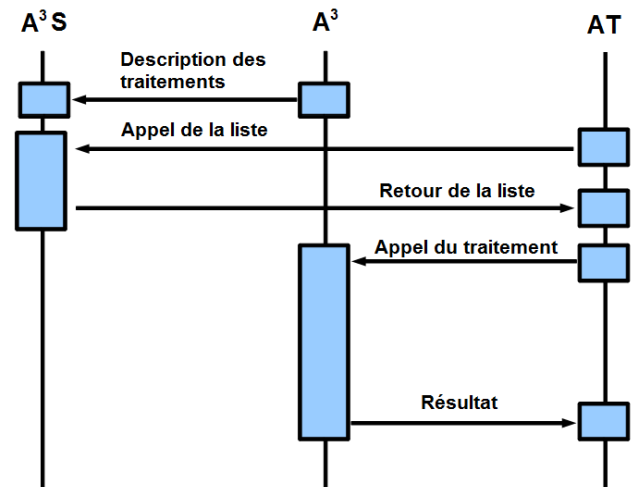


FIGURE 3: Séquence de requêtes

Le système proposé est illustré en figure 2. L'AT est un client qui effectue des requêtes vers un A³, sur un serveur distant. Sachant que le nombre d'A³ et d'AT opérationnels peut varier au fil des nouveaux développements, un autre serveur, appelé Superviseur des A³s (A³S), est inclus dans le système. Il a pour rôle de gérer une liste à jour des A³s disponibles. Ainsi lorsqu'un AT nécessite une liste des A³s, il envoie une requête au superviseur. Une fois la description de l'A³ obtenue l'AT et l'A³ peuvent communiquer directement. Afin d'échanger des fichiers vidéos de façon simple et rapide entre l'AT et l'A³, on utilise un Serveur de Fichiers Vidéos dédié (*Video File Server : VFS*).

3 Interactions au sein du système

L'AT permet aisément aux utilisateurs de définir et d'exécuter de multiples requêtes, de façon supervisée. Il interagit avec toutes les composantes du système. Tout d'abord, il interroge le superviseur pour récupérer la liste des descriptions des A³s disponibles. L'AT peut envoyer la requête automatiquement à son lancement

ou sur demande de l'utilisateur. L'AT peut alors indiquer à l'utilisateur les descriptions des fonctions disponibles sur les divers A³s. Ensuite il communique avec l'A³ choisi pour traiter la vidéo. Lorsque l'utilisateur sélectionne une fonction, les paramètres de cette dernière sont définis par le remplissage des paramètres fournis par la description de l'A³ (cf. section 4). Une fois le processus terminé, l'A³ encapsule le résultat, sous forme d'un graphe d'annotation, et l'envoie en retour de la requête à l'AT.

Chaque A³ est considéré comme une entité implémentant diverses fonctions de traitement pour l'annotation. Pour faire appel à toutes ces fonctions, chaque fois qu'un A³ est ajouté, il transmet sa description au superviseur (figure 3). Cette description est un code XML contenant l'API et des informations telles qu'un identifiant unique, une adresse, le numéro du port réseau et un texte d'aide.

4 Format de description des données

Du fait de la diversité des structures de données des ATs, nous avons besoin d'une structure de données simple, globale et compatible. C'est pourquoi nous utilisons un format de graphe d'annotation basé sur celui défini dans Schmidt *et al.* Schmidt *et al.* (2008). Grâce à ce format, l'utilisateur peut facilement définir la séquence d'images à traiter et les paramètres correspondants. L'A³ doit lire ce graphe pour obtenir les paramètres nécessaires à son exécution et finalement y stocker les résultats. Les informations concernant la fonction à appeler et la ou les vidéos à traiter sont enregistrées dans les métadonnées associées aux graphes.

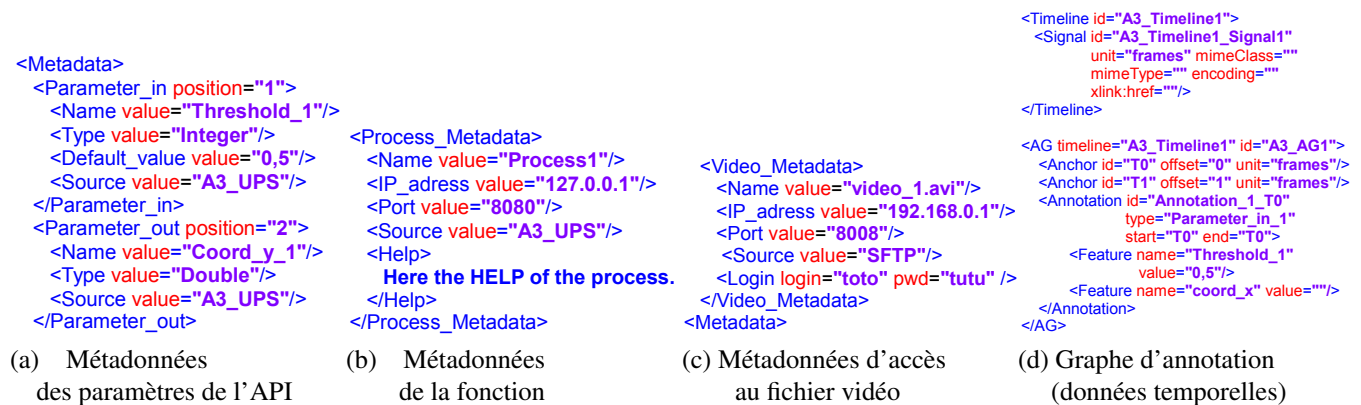


FIGURE 4: Format des données en XML

La figure 4 présente un exemple de structure de données encodé en XML. La partie 4a décrit un paramètre d'entrée et un paramètre de sortie au sein de l'API. Grâce au format XML, ces données peuvent être facilement analysées afin d'obtenir les différentes informations sur les paramètres, telles que leur type et leur valeur par défaut. Chaque paramètre utilise la balise `Parameters_in` ou `Parameters_out` avec un numéro de positionnement.

Les parties 4b et 4c décrivent les informations sur l'appel de la fonction et le traitement réalisé ainsi que l'accès au fichier vidéo sur le serveur, notamment des paramètres d'identification.

Finalement, la partie 4d est une utilisation classique des graphes d'annotation avec la définition de deux ancres temporelles et une annotation intercalée. Ce graphe contient les paramètres d'entrée accompagnés de leur valeur par défaut.

5 Développement logiciel

Pour réaliser cette architecture réseau, nous avons besoin d'une bibliothèque adaptée sachant que le système est multiplateforme et multilingage. Ainsi nous excluons les bibliothèques propriétaires ou propres à un langage telles que Java RMI ou Twisted. Pour réaliser ce type de système, on trouve deux sortes de bibliothèques : les *middleware* (ICE (ZeroC, URL), CORBA (ObjectManagementGroup, URL)) et les *Webservices* (SOAP (W3C, URL), XML, RPC (XML-RPC, URL)). La première est basée sur l'utilisation d'objets et d'appel de méthodes sur ces objets, tandis que la deuxième est basée sur l'utilisation de messages envoyés à des URL. Il est à noter que chacune de ces technologies répond à nos besoins avec plus ou moins de complexité. Nous avons décidé d'utiliser SOAP pour le prototype de cette architecture car son implémentation est aisée et il est totalement libre de droits.

6 Sécurité

Cette architecture nécessite également un système de sécurité. En effet, les vidéos utilisées ne sont pas forcément en accès public. Pour implémenter un niveau de sécurité suffisant, nous proposons deux composantes : l'utilisation d'HTTPS en tant que protocole de transfert sécurisé pour transmettre les données par SOAP et du protocole SFTP pour le transfert des vidéos entre le serveur vidéo et l'A³ ou l'AT.

7 Conclusion

En conclusion, nous proposons un système de communication permettant d'ajouter et d'utiliser aisément des fonctions de traitement automatique pour l'aide à l'annotation dans les outils d'annotation vidéo existants. Nous avons également défini un modèle de description des données échangées qui s'adapte aux formats des structures de données utilisées dans ces outils d'annotation. Ce modèle contient les métadonnées concernant l'appel des fonctions, l'accès au fichier vidéo et l'utilisation des paramètres d'entrée-sortie.

Le système est composé de quatre parties, situées dans des environnements différents : l'outil d'annotation (AT), les assistants automatiques d'annotation (A³), le superviseur des A³s (A³S) et le serveur des vidéos (VFS). Ainsi, la spécification et l'utilisation de SOAP pour le développement permettent une interopérabilité sécurisée, multiplateforme et multilingage (notamment RealBasic, C/C++ et Java).

Lors de futurs travaux, nous traiterons de l'ajout d'une communication asynchrone entre l'AT et les A³s pour éviter à l'utilisateur d'attendre les résultats lors de l'appel à de longs traitements. Aussi, nous souhaiterions synchroniser l'AT avec des applications interactives comme *Signing Avatar Synthesizer* (Kennaway *et al.*, 2007) ou *Signing Space Annotation Tool* (Lenseigne et Dalle, 2005). De plus, nous évaluerons la contribution des annotations automatiques pour les tâches d'annotation.

Remerciements

Les recherches amenant à ces résultats ont reçu le financement du 7ème programme-cadre Communauté Européenne (FP7/2007-2013) sous l'accord n° 231135.

Références

- BIRD S., LIBERMAN M. (2001). A formal framework for linguistic annotation. *Speech Communication*, **33**(No 1-2), 23–60.
- BRAFFORT A., CHOISIER A., COLLET C., DALLE P., GIANNI F., LENSEIGNE B., SEGOUAT J. (2004). Toward an annotation software for video of sign language, including image processing tools and signing space modelling. In *Proc. of 4th International Conference on Language Resources and Evaluation - LREC 2004*, volume 1, p. 201–203, Lisbon, Portugal.
- HANKE T. (2002). ILEX - a tool for sign language lexicography and corpus analysis. In *Proc. of 3rd International Conference on Language Resources and Evaluation, LREC 2002*, p. 923–926, Las Palmas de Gran Canaria, Spain.
- HANKE T., STORZ J. (2008). ILEX - a database tool for integrating sign language corpus linguistics and sign language lexicography. In *Proc. of 6th International Conference on Language Resources and Evaluation, LREC 2008*, p. W25–64–W25–67, Marrakesh.
- KENNAWAY J., GLAUERT J., ZWITSERLOOD I. (2007). Providing signed content on the internet by synthesized animation. *ACM Transactions on Computer-Human Interaction*, **14**(3), 15/1–19.
- KIPP M. (2001). Anvil - a generic annotation tool for multimodal dialogue. In *Proc. of 7th European Conference on Speech Communication and Technology (Eurospeech)*, p. 1367–1370.
- LEFEBVRE-ALBARET F., DALLE P. (2009). Body posture estimation in a sign language video. In *Proc of The 8th International Gesture Workshop*.
- LENSEIGNE B., DALLE P. (2005). Using signing space as a representation for sign language processing. In *Proc. of 6th International Gesture Workshop - GW 2005*, p. 25–36, Berder Island, France : Springer-Verlag.
- OBJECTMANAGEMENTGROUP (URL). Corba documentation. <http://www.omg.org/technology/>.
- ONG S., RANGANATH S. (2005). Automatic sign language analysis : A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 873–891.
- SCHMIDT T., DUNCAN S., EHMER O., HOYT J., KIPP M., LOEHR D., MAGNUSSON M., ROSE T., SLOETJES H. (2008). An exchange format for multimodal annotations. In *Proceedings of the 6th International Language Resources and Evaluation (LREC'08)*, p. 207–221, Marrakech, Morocco : European Language Resources Association (ELRA). <http://www.lrec-conf.org/proceedings/lrec2008/>.
- THEODORAKIS S., KATSAMANIS A., MARAGOS P. (2009). Product-HMMs for automatic sign language recognition. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 1601–1604 : IEEE Computer Society.
- W3C (URL). Soap documentation. <http://www.w3.org/TR/soap/>.
- WITTENBURG P., BRUGMAN H., RUSSEL A., KLASSMANN A., SLOETJES H. (2006). Elan : A professional framework for multimodality research. In *Proc. of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, p. 1556–1559.
- XML-RPC (URL). Xml-rpc documentation. <http://www.xmlrpc.com/spec>.
- ZEROC (URL). Ice documentation. <http://www.zeroc.com/download/Ice/3.4/Ice-3.4.0.pdf>.