



HAL
open science

AgentSlang: une Plateforme pour Systèmes Interactifs Distribués Multi-Incarnation

Sandratra Rasendrasoa, Maël Bouabdelli, Alexandre Pauchet, Julien Saunier, Sébastien Adam

► **To cite this version:**

Sandratra Rasendrasoa, Maël Bouabdelli, Alexandre Pauchet, Julien Saunier, Sébastien Adam. AgentSlang: une Plateforme pour Systèmes Interactifs Distribués Multi-Incarnation. Workshop sur les Affects, Compagnons artificiels et Interactions, Jun 2020, Saint Pierre d'Oléron, France. hal-02933465v1

HAL Id: hal-02933465

<https://inria.hal.science/hal-02933465v1>

Submitted on 8 Sep 2020 (v1), last revised 25 Sep 2020 (v2)

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.

AgentSlang: une Plateforme pour Systèmes Interactifs Distribués Multi-Incarnation

Sandratra Rasendrasoa² Maël Bouabdelli¹
Alexandre Pauchet¹ Julien Saunier¹ Sebastien Adam²
1-prénom.nom@insa-rouen.fr
2-prénom.nom@univ-rouen.fr

Normandie Université, INSA Rouen-Normandie, Université de Rouen-Normandie, LITIS

ABSTRACT

Les systèmes interactifs incarnés, virtuels ou robotiques, sont de plus en plus riches en terme de capacités dialogiques et affectives, ainsi que de comportements multimodaux. Cependant, cette richesse rend difficile la conception de nouveaux systèmes. Dans ce cadre, une approche par composants permet d'une part de réutiliser ceux-ci dans différents contextes, et d'autre part de répartir les calculs. Dans cet article, une démonstration des nouvelles fonctionnalités de la plateforme AgentSlang est présentée. Les améliorations apportées sont principalement de deux ordres: (1) l'intégration de mécanismes de dialogue fondés sur des modèles issus de l'apprentissage automatique, et (2) le support des agents robotiques. Ces ajouts permettent d'améliorer le potentiel de prototypage rapide d'AgentSlang, notamment pour étudier un modèle d'interaction au travers de plusieurs types d'incarnation.

KEYWORDS

Intelligence artificielle, interaction humain-agent, interaction humain-robot, système d'interaction distribué

1 INTRODUCTION

L'interaction humain-agent nécessite de nombreuses fonctionnalités: reconnaissance vocale, traitement du langage, la gestion du dialogue, génération de réponses, etc. Chacune de ces tâches est difficile mais peut être traitée séparément, ce qui implique de développer et réutiliser des briques logicielles. Afin de pouvoir réutiliser ces différentes briques fonctionnelles, il est nécessaire de faire abstraction au maximum de leurs architectures logicielles respectives. Une approche par composants est donc à privilégier pour ce type de systèmes. Par ailleurs, cette approche permet également d'aisément répartir la puissance de calcul. L'objectif de ces travaux est ainsi de proposer un système interactif capable de traiter de bout en bout une interaction humain-compagnon incarné, quelle que soit cette incarnation (robot ou agent virtuel).

2 LA PLATEFORME AGENTSLANG

AgentSlang¹ est une plateforme permettant de développer rapidement des Systèmes Interactifs Distribués [11]. Conçue dans le respect d'une approche orientée composant, la structure modulaire d'AgentSlang permet d'assurer l'intégration des différents services internes au système, via des composants réutilisables dédiés à chaque fonctionnalité du système.

D'autres solutions ont été proposées pour développer des systèmes interactifs. En particulier, ICT Virtual Human Toolkit (VH-Toolkit)² [5] et GRETA [9] sont deux systèmes interactifs dédiés à la conception d'agents conversationnels animés. VHToolkit propose une collection de composants pour la majorité des tâches nécessaires système interactif, telles que la reconnaissance et la synthèse vocale, la gestion du dialogue et des comportements non verbaux. GRETA propose une architecture développée en Java permettant d'interagir avec un personnage virtuel du même nom. La plateforme est conçue pour être utilisée principalement sur Windows avec Java 1.8. Le rendu 3D peut être effectué à l'aide des moteurs graphiques Ogre ou Unity[9]. D'une part, VHToolkit est la plateforme la plus aboutie, mais son code fermé rend son utilisation contraignante et ne permet pas de modifier ses composants. D'autre part, GRETA est plus orientée sur l'étude des agents virtuels, en particulier dans leur composante d'animation.

AgentSlang propose ainsi une alternative open-source et indépendante de l'incarnation à ces deux plateformes. Son code est distribué sous license GNU LGPLv3 et CeCILL-B. Une liste non-exhaustive de ses fonctionnalités est présentée ci-après :

- Afin de traiter les entrées utilisateurs, plusieurs modules sont disponibles: (1) un module de transcription vocale automatique est réalisée via l'API Google Speech; (2) des composants permettant le traitement du langage naturel via Senna [2], un système versatile permettant notamment l'annotation grammaticale du discours, la segmentation du discours et la reconnaissance d'entités nommées.
- Pour la gestion du dialogue, la plateforme propose des composants réalisés à l'aide de Syn!bad [10], qui est un langage d'expression régulière basé sur des synonymes. Le modèle de dialogue proposé suit sur une approche déterministe : une correspondance est faite entre l'entrée utilisateur et un pattern réalisé via Syn!bad; une réponse est ensuite générée en fonction du motif identifié.
- La plateforme permet aussi de traiter l'aspect émotionnel de l'interaction, via un composant de détection de la valence émotionnelle [11].
- Il est enfin possible de générer des comportements et les réponses à une requête utilisateur, par l'intermédiaire d'un agent virtuel, via des composants: (1) qui traitent la synthèse vocale en employant le logiciel Cereproc Voice ³, et (2) qui génèrent et transmettent un fichier BML transmis à un outil de représentation du comportement compatible avec ce format tel que MARC [3] ou GRETA[9].

¹<https://github.com/AgentSlang/agentslang.github.io>

²<https://vh toolkit.ict.usc.edu/>

³<https://www.cereproc.com/fr/products/voices>

3 NOUVEAUX COMPOSANTS

De nouveaux composants ont été intégrés à la plateforme de façon à (1) proposer une nouvelle approche concernant les mécanismes de dialogue, (2) assurer l'interfaçage avec des agents robotiques.

3.1 Mécanismes de dialogue

Trois fonctionnalités ont été intégrées à la plateforme pour enrichir la gestion du dialogue: (1) Un composant pour la détection automatique de l'interlocuteur dans un contexte multi-partie a été proposée par [7]. (2) Un composant d'annotation automatique des actes dialogiques a également été ajouté [8] pour faciliter l'interprétation des entrées utilisateurs. (3) Un composant fondé sur le système DrQA [1] et le chatbot proposé par [4] a été intégré pour traiter deux cas d'usage: le dialogue ouvert et le système de question réponse à partir de transcriptions vocales de l'utilisateur.

Ces nouveaux composants facilitent l'extraction d'informations pertinentes et la gestion du dialogue à partir des entrées utilisateurs, et d'autre part permettent aux utilisateurs d'interagir sur des sujets plus divers avec les agents interfacés à la plateforme.

3.2 Gestion des robots

Les composants de la plateforme AgentSlang [11] ont été initialement conçus pour de l'interaction humain-agent virtuel. Afin d'intégrer la possibilité de communiquer avec des robots, les composants nécessaires ont été ajoutés. Ainsi, du fait que ROS⁴ est le middleware le plus utilisé en robotique, des noeuds ROS ont été développés pour gérer la communication avec AgentSlang.

Les communications entre composants AgentSlang s'effectuent via la bibliothèque de messagerie asynchrone ZeroMQ⁵ [6]. Cette librairie haut niveau a été comparée avec d'autres solutions et présente deux avantages: elle est plus efficace, et elle supporte plusieurs types de sockets [11]. De plus, cette bibliothèque a pour avantage de disposer d'implémentations dans la majorité des langages de programmation classique, ce qui permet de communiquer entre différents programmes tout en faisant abstraction de leur langage. Pour les motifs évoqués plus haut mais aussi pour des raisons d'uniformité, cette API est utilisée pour les communications entre AgentSlang et ROS, à travers un mécanisme de publish/subscribe.

Afin de ne pas disséminer la localisation des composants entre noeuds ROS et composants AgentSlang, ainsi que de permettre une réutilisation dans le cas d'incarnations par agents virtuels, l'approche de conception a pour objectif de centrer les tâches de traitement dialogique génériques au sein d'AgentSlang. Les applications agent virtuel ou robot partagent donc une base commune et seule la partie émission et réception des informations varie selon l'incarnation. Cette architecture est présentée sur la figure 1.

4 CONCLUSION

Dans cet article de démonstration, nous présentons les évolutions récentes de la plateforme AgentSlang, qui permettent maintenant aux utilisateurs d'effectuer des discussions ouvertes grâce à de nouveaux composants. De plus, il est désormais possible d'effectuer aisément des interactions à l'aide d'incarnations robotiques par

réutilisation des mêmes composants décisionnels que ceux utilisés dans le cas d'une interaction humain-agent virtuel.

Ces nouvelles fonctionnalités facilitent la réalisation d'expérimentations. Dans un premier temps, il serait intéressant de comparer les nouveaux modèles de langue avec ceux initialement inclus. Dans un second temps, l'impact de l'incarnation sur l'interaction humain-agent selon différents contextes pourra être mesuré.

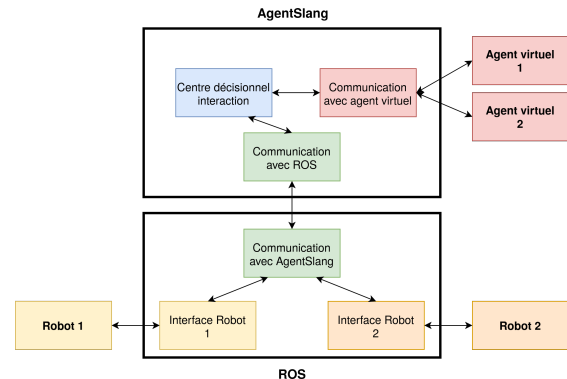


Figure 1: Schéma simplifié qui présente le support multi-incarnation de AgentSlang

REFERENCES

- [1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051* (2017).
- [2] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research* 12, Aug (2011), 2493–2537.
- [3] Matthieu Courgeon, Jean-Claude Martin, and Christian Jacquemin. 2008. User's gestural exploration of different virtual agents' expressive profiles. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 1237–1240.
- [4] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from Dialogue after Deployment: Feed Yourself, Chatbot! *arXiv preprint arXiv:1901.05415* (2019).
- [5] Arno Hartholt, David Traum, Stacy C Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. 2013. All together now. In *International Workshop on Intelligent Virtual Agents*. Springer, 368–381.
- [6] Peter Hintjens. 2013. *ZeroMQ: Messaging for Many Applications*. O'Reilly Media.
- [7] Usman Malik, Mukesh Barange, Naser Ghannad, Julien Saunier, and Alexandre Pauchet. 2019. A Generic Machine Learning Based Approach for Addressee Detection In Multiparty Interaction. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*. 119–126.
- [8] Usman Malik, Mukesh Barange, Julien Saunier, and Alexandre Pauchet. 2018. Performance Comparison of Machine Learning Models Trained on Manual vs ASR Transcriptions for Dialogue Act Annotation. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 1013–1017.
- [9] Radoslaw Niewiadomski, Elisabetta Bevacqua, Maurizio Mancini, and Catherine Pelachaud. 2009. Greta: an interactive expressive ECA system. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. 1399–1400.
- [10] Ovidiu Șerban. 2014. Syn! Bad: A Synonym-based Regular Expression Extension For Knowledge Extraction Task. *Studia Universitatis Babeș-Bolyai, Informatica* 59, 1 (2014).
- [11] Ovidiu Șerban and Alexandre Pauchet. 2014. AgentSlang: A New Distributed Interactive System - Current Approaches and Performance. In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART, INSTICC, SciTePress*, 596–603. <https://doi.org/10.5220/0004907305960603>

⁴<https://www.ros.org/>

⁵<https://zeromq.org/>