



HAL
open science

Construction d'agents artificiels flexibles et intelligibles afin d'évaluer et contrôler la qualité logicielle en collaboration avec l'équipe de développement humaine

Sébastien Bertrand, Pierre-Alexandre Favier, Jean-Marc André

► To cite this version:

Sébastien Bertrand, Pierre-Alexandre Favier, Jean-Marc André. Construction d'agents artificiels flexibles et intelligibles afin d'évaluer et contrôler la qualité logicielle en collaboration avec l'équipe de développement humaine. 2019. hal-04574973

HAL Id: hal-04574973

<https://hal.science/hal-04574973>

Preprint submitted on 14 May 2024

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.



Distributed under a Creative Commons Attribution 4.0 International License

CONSTRUCTION D'AGENTS ARTIFICIELS FLEXIBLES ET INTELLIGIBLES AFIN D'ÉVALUER ET CONTRÔLER LA QUALITÉ LOGICIELLE EN COLLABORATION AVEC L'ÉQUIPE DE DÉVELOPPEMENT HUMAINE

PRÉPUBLICATION

Sébastien Bertrand

onpoint
Bordeaux, France
s.bertrand@groupeonpoint.com

Pierre-Alexandre Favier

École Nationale Supérieure de Cognitique
IMS Laboratory, UMR 5218, ENSC/IPB
Bordeaux, France
pierre-alexandre.favier@ensc.fr

Jean-Marc André

École Nationale Supérieure de Cognitique
IMS Laboratory, UMR 5218, ENSC/IPB
Bordeaux, France
jean-marc.andre@ensc.fr

Décembre 2019

RÉSUMÉ

Ce papier présente un sujet de doctorat portant sur la conception d'agents artificiels, avec pour objectif de maximiser la pertinence dans le cadre d'une collaboration avec une équipe humaine. Il s'agira d'exploiter les travaux existants sur l'explicabilité en intelligence artificielle et sur la robotique développementale afin de créer des agents artificiels plus flexibles et transparents. Il faudra tenir compte des processus de prise de décision chez l'humain, seul ou en équipe, et des dynamiques de connaissance qui peuvent exister au sein d'une équipe. Le cas d'application sera l'évaluation et la mise sous contrôle de la qualité du code d'un artefact logiciel en collaboration avec l'équipe de développement.

Keywords Agent artificiel · Intelligence artificielle · Explicabilité · Développement · Collaboration humain-machine

1 Contexte

Onepoint est une entreprise de services du numérique (ESN), mais aussi, dans le sud-ouest, un éditeur de serious games. Elle produit essentiellement du logiciel, et les codes sources de ces derniers sont centraux dans son activité.

Son approche est holistique et porte un intérêt à tout le processus de création logiciel, depuis l'étude du besoin et la conception centrée utilisateur, avec par exemple la conception créative (design thinking, [17]), jusqu'à la réalisation et le contrôle qualité en passant par l'organisation des équipes de production, avec par exemple les méthodes agiles. Son approche est également transverse puisqu'elle intègre des problèmes sociétaux plus larges qui touchent nos clients et nos collaborateurs, comme le bien-être au travail et l'écologie.

Une entreprise comme onepoint intervient sur des bases de code qui varient tant en termes de technologie qu'en termes de conception et d'organisation interne du code, certaines étant maîtrisées de bout en bout, et d'autres parfois faiblement maîtrisées par les équipes amenées à intervenir dessus.

L'évaluation de la qualité entrante et sortante d'un logiciel, à savoir, sa conformité, sa maintenabilité et sa lisibilité, revêt donc un intérêt crucial pour toutes les activités de développement. Elle se fait principalement selon les approches suivantes :

- l'analyse statique du code, qui permet d'appliquer un certain nombre de contrôles automatiques en se basant sur un jeu de règles existant ou propres au contexte ;
- les tests dynamiques automatisés, qui vont contrôler le logiciel lors de son exécution, et vérifier qu'il se comporte comme attendu ;
- les revues et tests manuels par des pairs, souvent experts sur la technologie et/ou sur le domaine métier concerné.

Ces approches souffrent toutefois de plusieurs difficultés dans leur exécution. Elle sont longues à implémenter et nécessitent d'être adaptées à chaque contexte projet, sans toutefois être réutilisables sur d'autre projet. De plus elles sont facilement contournables par des développeurs qui peuvent adapter le code, afin qu'il respecte à moindre effort mais artificiellement les règles de qualité édictées sur le projet, sans pour autant respecter « l'esprit de la règle », ceci pouvant mener paradoxalement à une qualité dégradée.

La qualité logicielle, ainsi que son évaluation pertinente et efficiente, jouent un rôle central dans toute l'activité de onepoint. Elle permet de procéder à des estimations de coûts beaucoup plus précises, de fournir des livrables logiciels de qualité plus homogène et plus élevée, et par conséquent de limiter les défauts logiciels issus d'une mauvaise implémentation, favorisant in fine la compétitivité en limitant les coûts des développements logiciel.

2 Objectif

2.1 Enoncé de l'objectif

L'objectif de ce sujet de doctorat est de développer un agent artificiel, exploitant les dernières avancées de l'intelligence artificielle, afin d'évaluer et de mettre sous contrôle la qualité d'un artefact logiciel. Pour répondre aux contraintes industrielles propre à l'ingénierie logicielle, cet agent devra être facile à mettre en œuvre, portable, flexible, intelligible et pédagogique.

La facilité de mise en œuvre est essentielle afin de limiter les coûts au démarrage sur un projet, sans quoi l'adoption et l'exploitation effective de l'agent serait freinée voir exclue sur les projets.

Le portabilité permettra de profiter des acquis de projets similaires précédents, afin de mutualiser les coûts et d'augmenter la cohérence des critères de qualité entre les projets.

La flexibilité sera indispensable afin que l'agent puisse être adapté efficacement et facilement, sur des projets dont les contraintes sont éloignées, évoluant au cours de la vie ces derniers et en fonction des différents développeurs au sein des équipes.

L'intelligibilité sera primordiale lors de l'interaction des développeurs avec l'agent, dont la logique interne devra leur être accessible et compréhensible, pour qu'ils puissent juger facilement de la pertinence des remontées de l'agent, et les corriger au besoin.

L'aspect pédagogique permettra de favoriser la montée en compétence des développeur moins expérimentés dans un référentiel établi par les experts mais porté par l'agent.

2.2 Pourquoi un agent ?

Comme présenté par [7], un agent est essentiellement "quelque chose qui agit", sur cette base les auteurs formalisent une définition d'un *agent autonome* :

Un agent autonome est un système qui fait parti d'un environnement, qui perçoit cet environnement et agit sur lui au cours du temps, poursuivant son propre agenda, influençant ainsi ce qu'il percevra dans le futur.

Les auteurs fournissent quelques exemples caricaturaux satisfaisants cette définition, à savoir un humain, une bactérie ou même le thermostat régulant la température d'une pièce. Par ailleurs, les logiciels classiques ne satisfont pas cette définition, car leurs actions (sorties) ne sont pas constatées au cours du temps (ils attendent des entrées) et n'affectent pas ce qu'il perçoivent (entrées) dans le futur.

Au vu des contraintes et des objectifs fixés, qui forment un environnement complexe, ouvert et dynamique, la mise en oeuvre d'agents paraît particulièrement adaptée à la problématique.

3 Verrous scientifiques

3.1 Comment mesurer la qualité d'un code source

La qualité logicielle est intrinsèquement multi-modale [4], et son évaluation a nécessité la mise en place de normes (eg ISO 9126), qui ont beaucoup évolué au fil du temps [10]. La littérature parle également de modèles de qualité logiciel, qui définissent un ensemble structuré de caractéristiques.

Nous nous intéressons ici plus précisément à la maintenabilité, qui se sous-divise en lisibilité, modifiabilité, stabilité et testabilité [4, 2], et qui est évalué à l'aide de métrique facilement mesurable, qui sont par exemple le volume, la complexité cyclomatique, ou la duplication de code, à l'aide d'un modèle de maintenabilité [2]. Les métriques sont également un champ d'étude très actif, qui s'appuie sur pléthores de résultats empiriques [13], mais nécessite de se confronter davantage aux problématiques industrielles.

De plus, dans [15], les auteurs mettent en évidence que les métriques objectives et les évaluations subjective par les développeurs ne convergent pas systématiquement dans les cas les plus complexes, ils proposent que la meilleure alternative consiste à utiliser les deux méthodes conjointement.

Une autre problématique concerne l'agrégation des métriques [3], afin de déterminer la maintenabilité de composants logiciels de plus haut niveau, en effet la maintenance effective d'un logiciel nécessite de prioriser les efforts de l'équipe de développement. Les résultats du modèle de [3] ont été corrélé significativement avec les évaluations des experts, toutefois une approche agent pourrait introduire plus d'extensibilité et de flexibilité au modèle utilisé.

Enfin les tests peuvent également nous renseigner sur la qualité du code [1], mais les résultats semblent paradoxaux car c'est le nombre d'objets évalués qui est corrélé positivement avec la complexité, or une équipe de développement ayant pour habitude d'effectuer des assertions minimalistes serait performante au vu de ce critère. Il y a clairement une concurrence entre la qualité des tests et la qualité du code de production, des agents ayant des objectifs d'évaluation différents seraient un moyen efficaces de modéliser cette concurrence.

3.2 Comment concevoir et exploiter une intelligence artificielle flexible ?

Le développement et l'exploitation de produits issus des travaux de l'intelligence artificielle rencontrent beaucoup de difficultés dans un contexte industriel, ceci malgré des résultats impressionnants obtenus par ailleurs sur certaines tâches. Selon Gartner[22] les projets d'application de l'intelligence artificielle sont encore aujourd'hui à 80% de « l'alchimie dirigés par des sorciers » et seulement 15% seront des succès.

En effet de nombreux facteurs limitent l'exploitation de tels produits. En premier lieu le découplage complet qui existe entre les phases d'entraînement et d'exploitation impose de reprendre systématiquement l'entraînement en cas de détection d'un biais, d'une anomalie ou s'il y a un besoin d'évolution, dans un contexte industriel cela oblige également à réexécuter en partie la recette du produit. Par ailleurs l'approche « boîte noire » des réseaux de neurones profond est de plus en plus remise en question, d'abord d'un point de vue réglementaire, avec par exemple le référentiel général de protection des données (RGPD), mais également en pratique puisque l'impossibilité d'accéder aux raisonnements constitue une limite forte à la validation, l'amélioration et l'acceptabilité de ces systèmes. Finalement le risque associé à la mise en place de produits, auxquels il est difficile d'appliquer les méthodes itératives propre aux projets informatiques modernes, ne favorise pas l'expérimentation et le prototypage industriel.

En génie logiciel, c'est la modularité des programmes qui permet de résoudre cette problématique, c'est à dire la décomposition d'un programme en composants relativement indépendants les uns des autres. L'approche des systèmes multi-agents correspond à la transposition naturelle du concept de modularité dans l'intelligence artificielle, et les applications faisant collaborer des agents artificiels avec des humains existent déjà. Ainsi, l'encyclopédie Wikipédia est largement maintenue par des agents artificiels[5, 8, 18, 16] dont le rôle consiste majoritairement dans le traitement en masse de tâches fastidieuses ou répétitives. Sur la version francophone de l'encyclopédie, il existe près d'une centaine de bots[14] qui travaillent quotidiennement en collaboration avec les mainteneurs. Par exemple, le célèbre Salebot[12, 20] en service depuis 2007, a été construit pour nettoyer le vandalisme sur les pages de Wikipédia.

Toutefois, ces agents artificiels prennent leurs décisions en se basant sur un système expert, c'est à dire une base de règles qui a été définie par l'étude précise du domaine d'application ciblé. L'exploitation des techniques récentes issues de l'intelligence artificielle permettrait de nouveaux cas d'usage intéressants où le système de prise de décision de l'agent se construirait et évoluerait en fonction de ses expériences.

C'est dans ce contexte du développement d'un agent artificiel autonome, exploitant des travaux sur l'apprentissage machine et les réseaux de neurones profonds, que l'explicabilité de leurs apprentissages pourra permettre d'optimiser et de démocratiser l'utilisation de systèmes basés sur l'intelligence artificielle. La prise en compte de la robotique développementale[19] permettra de développer des systèmes flexibles évoluant favorablement au cours du temps, au contact des humains collaborant avec l'agent. Finalement la prise en compte des connaissances concernant la prise de décision chez l'humain, seul ou en équipe, permettra de maximiser la pertinence d'un tel agent artificiel.

3.3 Qu'est ce qu'une bonne explication ?

Il existe beaucoup de travaux concernant l'explicabilité des boîtes noires en générale [11] et des réseaux de neurones en particuliers [9]. La différence entre ces approches étant le type d'inspections que l'on s'autorise à faire sur le système à expliquer. Les classifications sont riches et orthogonales :

- [9] répartie les méthodes en trois catégories : l'explication des processus, l'explication des représentations, ou les modèles auto-explicatif;
- [11] divise d'abord l'espace du problème, en considérant les explications globales, locales, l'inspection et les modèles boîtes blanches, puis s'intéresse aux types d'explications données et aux boîtes noires traitées, une attention particulière est apportée aux modèles généralisables, qui peuvent modéliser n'importe quel système.

Par ailleurs, si l'on s'intéresse aux explications en elles-mêmes, on peut les considérer comme des interfaces entre les humains et le système qui décide [11], et elles doivent donc constituer un bon modèle intermédiaire de ce dernier, c'est la complétude, tout en ayant du sens pour les humains, c'est l'interprétabilité. Il existe une compétition entre ces deux concepts, l'explication la plus simple donnant lieu à une approximation grossière du système, alors qu'une explication très détaillée serait incompréhensible[9]. Il serait également intéressant de considérer d'interagir sur l'explication en elle-même, en envisageant la production d'une justification, pour convaincre, plutôt qu'une explication, pour faire comprendre.

L'analyse et la catégorisation de la qualité des explications [6] montre que les arbres de décisions et les jeux de règles sont les plus efficaces, sans toutefois être très satisfaisants. Les approches locales, telle que LIME [21], peuvent permettre de fournir des explications plus facilement compréhensibles, mais valables uniquement sur l'exemple expliqué.

4 Perspectives d'applications

Les récents développements de l'automatisation des processus robotisés[23] montrent qu'il existe encore de nombreuses possibilités à explorer pour accompagner ou remplacer les travailleurs sur les tâches les plus ingrates. Le développement d'agents artificiels flexibles et intelligibles correspond à une approche intermédiaire dans le continuum d'automatisation des services. Les applications iront de l'analyse et du pré-traitement des entrants de la relation clients (courriels, chat, activité) à la surveillance prédictive automatisée d'un système informatique en production.

Concernant l'exploitation des travaux issus de l'intelligence artificielle, l'approche avec des agents portables et faciles à mettre en oeuvre, associant une équipe maître de la « formation » de ces derniers, permettra un souplesse opérationnelle bien meilleure que l'approche classique en deux temps. L'évaluation de la qualité d'un code source entre dans ce cadre, et l'écriture automatique de tests en est une continuité possible. D'autres applications similaires seront l'évaluation de l'accessibilité d'un site web, ou la vérification qu'une spécification fonctionnelle est valide et cohérente.

Enfin une approche avec des agents flexibles et pédagogiques permettra de nombreux usages éducatifs. L'identification d'experts ou de professeurs, ayant une interaction privilégiée avec les agents artificiels, permettra de faciliter le transfert d'expertise vers des collaborateurs moins expérimentés ou des élèves, en présentant de manière standardisée l'expertise

acquise. Les applications concrètes seront le nommage des différentes entités d'un code source (classe, méthode, etc), ou la sensibilisation à la sécurité des applications.

Références

- [1] Mauricio Finavaro ANICHE, Gustavo Analdi OLIVA et Marco Aurélio GEROSA. "What do the asserts in a unit test tell us about code quality? A study on open source and industrial projects". In : *Proceedings of the European Conference on Software Maintenance and Reengineering, CSMR March* (2013), p. 111-120. ISSN : 15345351. DOI : 10.1109/CSMR.2013.21.
- [2] Robert BAGGEN et al. "Standardized code quality benchmarking for improving software maintainability". In : *Software Quality Journal* 20.2 (2012), p. 287-307. ISSN : 15731367. DOI : 10.1007/s11219-011-9144-9.
- [3] Tibor BAKOTA et al. "A probabilistic software quality model". In : *IEEE International Conference on Software Maintenance, ICSM* (2011), p. 243-252. DOI : 10.1109/ICSM.2011.6080791.
- [4] B. W. BOEHM, J. R. BROWN et M. LIPOW. "Quantitative evaluation of software quality". In : *Proceedings - International Conference on Software Engineering*. 1976, p. 592-605.
- [5] Dominique CARDON et Julien LEVREL. "La vigilance participative. Une interprétation de la gouvernance de Wikipédia". In : *Réseaux 2* (2009), p. 51-89.
- [6] Hoa Khanh DAM, Truyen TRAN et Aditya GHOSE. "Explainable software analytics". In : *Proceedings - International Conference on Software Engineering* (2018), p. 53-56. ISSN : 02705257. DOI : 10.1145/3183399.3183424. arXiv : arXiv:1802.00603v1.
- [7] Stan FRANKLIN et Arthur C. GRAESSER. "Is it an Agent, or Just a Program? : A Taxonomy for Autonomous Agents". In : *Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings*. Sous la dir. de Jörg P. MÜLLER, Michael J. WOOLDRIDGE et Nicholas R. JENNINGS. T. 1193. Lecture Notes in Computer Science. Springer, 1996, p. 21-35. ISBN : 3-540-62507-0. DOI : 10.1007/BFb0013570. URL : <https://doi.org/10.1007/BFb0013570>.
- [8] R Stuart GEIGER et Aaron HALFAKER. "When the levee breaks : without bots, what happens to Wikipedia's quality control processes?" In : *Proceedings of the 9th International Symposium on Open Collaboration*. ACM, 2013, p. 6.
- [9] Leilani H. GILPIN et al. "Explaining Explanations : An Approach to Evaluating Interpretability of Machine Learning". In : *CoRR abs/1806.00069* (2018). arXiv : 1806.00069. URL : <http://arxiv.org/abs/1806.00069>.
- [10] Oleksandr GORDIEIEV et al. "Evolution of software quality models in context of the standard ISO 25010". In : *Advances in Intelligent Systems and Computing* 286 (2014), p. 223-232. ISSN : 21945357. DOI : 10.1007/978-3-319-07013-1_21. URL : http://link.springer.com/10.1007/978-3-319-07013-1_21.
- [11] Riccardo GUIDOTTI et al. "A Survey Of Methods For Explaining Black Box Models". In : *CoRR abs/1802.01933* (2018). arXiv : 1802.01933. URL : <http://arxiv.org/abs/1802.01933>.
- [12] Erwan JOUD, Nicolas JULLIEN et Marine LE GALL-ELY. "Le processus d'acceptation d'un bot : Analyse du récit de vie de Salebot". In : *32e Congrès International de l'AFM*. Lyon, France, mai 2016. URL : <https://hal.archives-ouvertes.fr/hal-01341906>.
- [13] Barbara KITCHENHAM. "What's up with software metrics? - A preliminary mapping study". In : *Journal of Systems and Software* 83.1 (2010), p. 37-51. ISSN : 01641212. DOI : 10.1016/j.jss.2009.06.041. URL : <http://dx.doi.org/10.1016/j.jss.2009.06.041>.
- [14] *Liste des bots francophone de Wikipédia*. <https://fr.wikipedia.org/wiki/Cat%C3%A9gorie:Wikip%C3%A9dia:Bot/Statut>.
- [15] Mika MÄNTYLÄ et Casper LASSENIUS. "Subjective evaluation of software evolvability using code smells : An empirical study". In : *Empirical Software Engineering* 11.3 (2006), p. 395-431. DOI : 10.1007/s10664-006-9002-8. URL : <https://doi.org/10.1007/s10664-006-9002-8>.
- [16] Daniel NASAW. *À la rencontre des « bots » qui veillent eux aussi sur Wikipédia*. <https://framablog.org/2012/09/17/bot-wikipedia/>. 2012.
- [17] Peter NEWMAN et al. "The Role of Design Thinking and Physical Prototyping in Social Software Engineering". In : *37th IEEE/ACM International Conference on Software Engineering, ICSE 2015, Florence, Italy, May 16-24, 2015, Volume 2*. Sous la dir. d'Antonia BERTOLINO, Gerardo CANFORA et Sebastian G. ELBAUM. IEEE Computer Society, 2015, p. 487-496. DOI : 10.1109/ICSE.2015.181. URL : <https://doi.org/10.1109/ICSE.2015.181>.

- [18] Thierry NOISSETTE. *Pour améliorer Wikipédia, une armée de robots trime en coulisse*. <https://www.nouvelobs.com/rue89/rue89-internet/20150216.RUE7882/pour-ameliorer-wikipedia-une-armee-de-robots-trime-en-coulisse.html>. 2015.
- [19] Pierre-Yves OUDEYER. “Autonomous development and learning in artificial intelligence and robotics : Scaling up deep learning to human-like learning”. In : *CoRR* abs/1712.01626 (2017). arXiv : 1712.01626. URL : <http://arxiv.org/abs/1712.01626>.
- [20] *Présentation du robot Salebot*. <https://fr.wikipedia.org/wiki/Utilisateur:Salebot>.
- [21] Marco Túlio RIBEIRO, Sameer SINGH et Carlos GUESTRIN. ““Why Should I Trust You ?” : Explaining the Predictions of Any Classifier”. In : *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. Sous la dir. de Balaji KRISHNAPURAM et al. ACM, 2016, p. 1135-1144. ISBN : 978-1-4503-4232-2. DOI : 10.1145/2939672.2939778. URL : <https://doi.org/10.1145/2939672.2939778>.
- [22] Andrew WHITE. *Our Top Data and Analytics Predicts for 2019*. https://blogs.gartner.com/andrew_white/2019/01/03/our-top-data-and-analytics-predicts-for-2019/.
- [23] Leslie WILLCOCKS et Mary LACITY. “The Outsourcing Unit Working Research Paper Series Robotic Process Automation : The Next Transformation Lever for Shared Services”. In : January (2016), p. 1-35. URL : www.lse.ac.uk/management/research/outsourcingunit.