Note d’opportunité

Sur la valorisation des logiciels issus de la recherche

Groupe Projet Logiciels Libres et Open Source du
Comité pour la Science Ouverte
Novembre 2019

Résumé

Le logiciel est un objet hybride au sein de la recherche, dont il est à la fois moteur (comme outil), résultat (comme preuve d’existence d’une solution) et objet d’études (comme artefact).

Ce statut spécifique doit amener à la définition de stratégies, d’outils et de procédures adaptés aux différents enjeux qu’il soulève, tels que notamment : la citation des contributions relatives à la conception et à la production de logiciels, la reproductibilité des résultats de recherche faisant intervenir des logiciels, la valorisation et la pérennisation du patrimoine logiciel créé.

Place du logiciel dans la recherche

La recherche a pour objet de produire des connaissances nouvelles, dans tous les domaines à la portée de l’esprit humain. Elle s’appuie sur la méthodologie scientifique, autrement dit sur la reproductibilité des résultats afin de garantir leur potentielle réfutation. La naissance de l’informatique, science du traitement efficace de l’information, a ouvert de nouvelles voies aux scientifiques. Tout comme les télescopes en leur temps, les ordinateurs ont permis d’augmenter le domaine de l’atteignable. Surtout, l’apparition du logiciel a permis de formaliser, sous une forme non ambigüe, des processus abstraits de traitement de l’information, afin qu’ils puissent éventuellement être mis en œuvre par des ordinateurs et partagés dans la communauté scientifique et au-delà, pour tous les citoyens.

Le logiciel joue donc dans la recherche un triple rôle :

1. il sert d’**outil** dans de nombreux domaines, en traitant efficacement divers types de données pour construire et tester des modèles visant à étayer ou invalider des hypothèses ;
2. il peut constituer en lui-même un **résultat** de recherche, en tant que preuve d’existence d’une solution algorithmique efficace à un problème donné, cette efficacité étant évaluée à l’aune des capacités des ordinateurs du moment ;
3. il peut être lui-même **objet** de recherche. En particulier, la communauté scientifique s’intéresse aux modes de développement des logiciels et à la preuve de leurs propriétés, en lien notamment avec les enjeux sociétaux liés à la transparence et à la confiance dans les traitements informatisés.

Il en découle que, de plus en plus, les scientifiques ne produisent plus seulement des articles de recherche synthétisant leurs résultats, mais aussi des logiciels venant en appui ou en démonstration de ceux-ci. Cette activité peut représenter une part importante de leur travail, qui doit être prise en compte de façon équitable dans leur évaluation par leurs pairs et tutelles.

Grâce au développement des réseaux numériques, ces logiciels sont, de plus en plus, construits de façon collaborative, soit par l’agrégation d’une communauté de contributeurs, soit par la réutilisation d’un nombre toujours croissant de briques logicielles elles-mêmes très souvent également construites de façon collaborative. La production logicielle moderne agrège donc des personnes aux multiples compétences, dont les contributions peuvent être de natures variées. Ainsi, un logiciel ne peut se

résumer à un ensemble d'ajouts historicisés de lignes de code (ou « *commits* »). La dynamique sous-jacente, impulsée par les différents architectes et leaders du projet scientifique de production logicielle, est une condition essentielle de sa réussite.

Ces conditions modernes de création logicielle influent fortement sur le statut juridique des œuvres logicielles produites. Ces spécificités doivent être prises en compte dans la définition de modèles de valorisation adaptés, permettant de maximiser l'impact sociétal, y compris en dehors du champ scientifique.

Enjeux

Les caractères spécifiques de la production logicielle dans le domaine de la recherche font émerger plusieurs enjeux :

1. Afin de maintenir les propriétés de reproductibilité et de réfutabilité des productions de recherche liées au logiciel, le simple exposé des résultats n'est en général plus suffisant. Il est nécessaire d'offrir à la communauté scientifique les moyens de **reproduire les conditions expérimentales** ayant conduit à leur obtention et d'éprouver les algorithmes proposés sur d'autres jeux de données. Garantir l'**accès pérenne aux logiciels comme aux données** qu'ils manipulent suppose de :
    a) pouvoir **faire référence de façon pérenne** à des versions particulières des logiciels utilisés ainsi que de leurs environnements d'exécution ;
    b) disposer de plateformes susceptibles de **conserver de façon pérenne** lesdites versions ;
    c) disposer d'environnements matériels et système permettant de ré-**exécuter à l'identique** les logiciels. Il s'agit d'un problème scientifique complexe, dans la mesure où l'obsolescence rapide des matériels peut avoir un impact fort sur la reproductibilité de certains types de résultats.
2. Afin que la visibilité et la réputation des chercheurs puissent refléter l'investissement d'une partie de leur temps dans la production logicielle, il est nécessaire de construire un mécanisme **de citation** adapté.
3. Pour mettre en œuvre une politique et des moyens permettant de **pérenniser et/ou valoriser de façon adéquate** les productions logicielles de la recherche publique, y compris en dehors du champ scientifique, il est nécessaire de disposer :
    a) de **méthodologies de référence** pour évaluer les différents modes de valorisation possibles, illustrées par des cas d'usages et des retours d'expériences ;
    b) d'un **inventaire** de ces productions, accessible de la façon la plus large possible.

Axes de travail

Sur l'archivage et le référencement

Pour ce qui concerne l'archivage pérenne des codes sources des logiciels, ainsi que le référencement précis des versions des codes sources pour des fins de traçabilité et reproductibilité scientifique, nous disposons aujourd'hui de solutions que l'on peut recommander à l'usage des chercheurs dans toutes les disciplines[1].

Sur le système de citation/réputation

Comme il a été décrit plus haut, un logiciel est le résultat d'un processus complexe mêlant activités de conception et de développement[2]. Il ne peut être réduit à une somme de lignes de code, car il évolue, pas plus qu'il ne peut être restreint à une somme de « *commits* », car la valeur des contributions ne se mesure pas au nombre de lignes produites[1]. Une contribution de nature architecturale ou algorithmique peut en effet ne pas apparaître directement en tant que production formelle de lignes de code, car les systèmes de gestion du code source ne rendent visibles que les noms des développeurs. La **création de traces relatives aux contributions** est donc tant un problème technique (moyens effectifs de citation) qu'organisationnel (moyens de matérialiser ces contributions au sein de l'environnement de développement), et nécessite en particulier des procédures de contrôle de la qualité des métadonnées, absentes dans des dépôts tels que FigShare ou Zenodo[3] et en cours de développement sur HAL[4].

Aussi, certains aspects du **statut juridique des productions logicielles issues de la recherche** ont besoin d'être clarifiés : quelles sont les interactions entre le droit moral des chercheurs (incessible et inaliénable) et la cession ou la dévolution des droits patrimoniaux (automatique lorsque les chercheurs sont employés d'un organisme public) ? Quels sont les critères pour reconnaître la qualité d'auteur aux contributeurs ? En particulier, une personne qui a contribué à un logiciel en définissant les modèles des problèmes à résoudre, en concevant les algorithmes, en définissant l'architecture du logiciel ou en dirigeant les travaux de développement, doit évidemment recevoir crédit pour cela ; mais peut-elle aussi être considérée comme un auteur au sens juridique du terme alors qu'elle n'a pas produit une seule ligne de code ? Comment la loi pour une République numérique s'applique-t-elle dans le cadre des productions logicielles de la recherche ?

Sur la valorisation des productions logicielles

En termes de référencement, les bases de données de logiciels produits par la recherche sont souvent des outils internes, mêlant les problématiques de référencement et d'évaluation interne des chercheurs, ce qui empêche de les ouvrir largement. Des tentatives de constitution de bases de données publiques ont été menées[5], mais cela conduit à dupliquer la saisie d'une partie des données, qui ne sont pas automatiquement mises à jour. Il est donc nécessaire de définir un **socle homogène de données ouvertes**, éventuellement complété de données à visée interne, qui ont tout intérêt à être également normalisées. La mise à disposition partagée de ce socle auprès des différents établissements académiques leur permettrait de recenser de façon unique le patrimoine logiciel issu de leurs agents et dont ils sont parfois co-titulaires des droits en indivision.

À l'heure actuelle, les méthodologies de valorisation des logiciels produits par la recherche ne sont pas uniformes. Il est donc nécessaire, après recensement, de **définir des méthodologies de valorisation de référence**, en s'appuyant sur les mécanismes déjà mis en œuvre (licences libres et/ou privatives, création de fondations ou de consortiums, etc.), et de les **partager auprès des acteurs concernés** (services de valorisation des établissements académiques et SATT).

1 Les contributions peuvent prendre la forme de demandes argumentées de nouvelles fonctionnalités, de rapports d'anomalies suite à usage dans un nouveau contexte scientifique, de portages sur de nouvelles plateformes, d'améliorations ergonomiques au niveau de l'interface, etc.

Sur la pérennisation du patrimoine logiciel issu de la recherche

Si la production de logiciels constitue une activité de recherche à part entière, il n'en va pas de même de leur maintenance[6]. Dès le moment où la question scientifique ayant motivé la production du logiciel est résolue, que le logiciel existant ne permet plus d'obtenir les nouveaux résultats espérés ou encore que les concepteurs s'orientent vers de nouveaux projets, rien ne garantit la pérennité du logiciel produit. Or, un logiciel qui n'est plus un objet de recherches peut néanmoins contribuer, en tant qu'outil, à l'obtention de résultats par d'autres équipes, voire être exploité par des entreprises comme outil de développement ou dans le cadre d'une exploitation industrielle ou commerciale. La question de la maintenance et de la pérennisation des logiciels issus de la recherche doit donc être anticipée, tant par les concepteurs que par les usagers[7].

Pour les usagers, il est essentiel de recenser l'ensemble des logiciels jouant un rôle stratégique dans leurs processus métiers et de s'assurer, en interrogeant les concepteurs, que les services de maintenance et éventuellement d'amélioration pourront être garantis. Pour les concepteurs, la connaissance de l'usage par différentes catégories d'acteurs, et sa criticité relative, doit permettre d'évaluer les ressources que les usagers seraient prêts à mobiliser afin de pérenniser le maintien en conditions opérationnelles du logiciel (portage sur des systèmes et environnements récents, gestion des dépendances avec des logiciels tiers, débogage) et son évolution.

Les concepteurs de logiciels ne sont souvent pas suffisamment outillés pour intégrer ces problématiques dans leur travail, et les dispositifs de soutien aux projets sont souvent peu connus. Ceux-ci peuvent être des mécanismes souples, tels que le recours à un consortium (voir les projets SSI[2] ou ReSA[3]) ou une fondation permettant de collecter des fonds et d'héberger des personnels dédiés, ou la mise à disposition directe de main d'œuvre (du temps d'ingénieur) par les structures de recherche. Ils peuvent également consister, en première intention ou en complément des dispositifs précédents, en la création d'une entreprise dédiée ou au transfert de l'édition à une société existante[4].

Sur la mutualisation des ressources

Comme plusieurs expériences passées l'ont montré (telles que le projet Depsy[5]), il n'est pas opérant de considérer la création des outils nécessaires aux finalités énoncées ci-dessus comme des projets de développement à financer de façon isolée. L'objectif doit être la création d'une **infrastructure** :

- **unique**, même si elle est basée sur une architecture distribuée hébergée localement au sein de multiples institutions, afin d'éviter toute dispersion des efforts ;

2 Le Software Security Institute vise, entre autres, à apporter une expertise en ingénierie logicielle aux concepteurs et mainteneurs de logiciels pour la recherche, dans le but d'accroître leur pérennité : https://www.software.ac.uk/

3 La Research Software Alliance (ReSA) est un groupement de personnes impliquées dans la production de logiciels pour la recherche, souhaitant voir cette activité pleinement reconnue sur le plan académique. Leur site recense notamment les avancées scientifiques explicitement rendues possibles par l'usage de logiciels : https://www.researchsoft.org/. Au Royaume-Uni, l'UK Research Software Engineer Association (UKRSE) poursuit des buts similaires : https://rse.ac.uk/

4 Comme par exemple le transfert à la société Kereval du développement de la plate-forme de test d'interopérabilité Gazelle.

5 Le projet Depsy a été doté d'un financement unique de 675 k$. Aucun moyen n'ayant été prévu pour sa pérennisation, la terminaison du projet a mécaniquement conduit à l'arrêt de sa maintenance et de son évolution.

- **pérenne** et **publique**, l'exemple de Google Code, plateforme de développement logiciel gratuite qui a été fermée en 2015, ayant démontré que le secteur privé ne peut garantir une pérennité sur le long terme.

Le portail d'accès qui pourrait être adjoint à cette infrastructure permettrait de lier, conceptuellement et fonctionnellement, les différentes finalités de préservation, catalogage, référencement et diffusion/valorisation, en s'adressant aux différents publics visés : les personnels académiques et industriels en recherche d'une solution à leurs besoins, ceux qui y contribuent ou souhaitent le faire, ceux qui explorent pour des raisons scientifiques les données entreposées, les tutelles, voire même le grand public.

Les travaux relatifs aux axes de travail présentés doivent s'inscrire dans un cadre pérenne d'allocation de moyens humains et financiers, seul à même de garantir le retour sur investissement des moyens engagés.

Recommandations

Les axes de travail ci-dessus peuvent être déclinés en un certain nombre de recommandations :

Recommandation n° 1 : Entrer dans la **conversation internationale** et susciter des **collaborations** sur le sujet.

Recommandation n° 2 : Faire reconnaître la **spécificité du logiciel**, qui n'est pas « juste une donnée », en particulier dans la discussion sur la notion de FAIR data.

Recommandation n° 3 : Promouvoir les bonnes pratiques pour **l'archivage et le référencement** des logiciels de recherche.

Recommandation n° 4 : Construire une **notion consensuelle** de ce qu'est une « **contribution** » à un logiciel de recherche.

Recommandation n° 5 : Construire **des outils** mettant en œuvre cette notion de contribution dans le but de pouvoir créditer effectivement des auteurs/concepteurs pour leurs contributions logicielles.

Recommandation n° 6 : Promouvoir un **schéma normalisé de métadonnées** partageables relatives aux logiciels, en vue d'une ouverture des métadonnées de logiciels issus de la recherche.

Recommandation n° 7 : Encourager les établissements académiques à partager les **métadonnées** des logiciels de recherche.

Recommandation n° 8 : Définir **une stratégie** et des **procédures communes** d'évaluation, de pérennisation et de valorisation des logiciels sous licences libres.

Recommandation n° 9 : Favoriser la création de « **boîtes à outils juridiques** » permettant de pérenniser les logiciels libres issus de la recherche.

Contact et Diffusion

Ce texte est diffusé sous la licence Creative Commons CC-BY 4.0.

Les auteurs peuvent être contactés en écrivant à roberto@dicosmo.org ou francois.pellegrini@labri.fr

1 Roberto Di Cosmo, How to use Software Heritage for archiving and referencing your source code: guidelines and walkthrough, https://www.softwareheritage.org/save-and-reference-research-software/

2 Pierre Alliez, Roberto Di Cosmo, Benjamin Guedj, Alain Girault, Mohand-Said Hacid, Arnaud Legrand, Nicolas P. Rougier, Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria, https://hal.archives-ouvertes.fr/hal-02135891v1

3 Zenodo. https://about.zenodo.org/policies/

4 Voir la description du dépôt modéré des logiciels de recherche dans Software Heritage via HAL : https://www.softwareheritage.org/2018/09/28/depositing-scientific-software-into-software-heritage/?lang=fr

5 Sophie Nicoud, Après PLUME : FENIX (Fiches d'Évaluation Normalisée Issues de l'expérience), https://resinfo.org/les-newsletters-de-resinfo/NewsLetter-3/Apres-PLUME-FENIX-Fiches-d-Evaluation-Normalises-Issues-de-l-eXperience

6 Anna Nowogrodzki, How to support open-source software and stay sane, Nature n° 571, pp. 133-134, juillet 2019, doi:10.1038/d41586-019-02046-0. https://www.nature.com/articles/d41586-019-02046-0

7 Dalmeet Singh Chawla, The unsung heroes of scientific software. Nature n° 529, pp. 115-116, janvier 2016, doi:10.1038/529115a. https://www.nature.com/news/the-unsung-heroes-of-scientific-software-1.19100