Le pilier logiciel de la science ouverte

Construire le pilier logiciel de la science ouverte

Roberto Di Cosmo
Software Heritage/Inria/Université de Paris

Merci beaucoup de m'accueillir ici aujourd'hui. C'est un réel plaisir d'ouvrir cette session sur la place des logiciels dans la science ouverte. Permettez-moi de commencer par vous présenter un peu le contexte. Si nous regardons autour de nous, nous voyons des logiciels partout. Ils alimentent notre industrie, nourrissent les innovations et sont essentiels à la recherche universitaire. Ils sont le tissu qui lie nos vies numériques et professionnelles. C'est grâce aux logiciels que nous sommes aujourd'hui en mesure de participer à cette conférence alors que nous sommes en pleine pandémie. Cependant, lorsque nous parlons de logiciels, nous oublions parfois qu'ils ne sortent pas du néant. Ce ne sont pas simplement des morceaux de données qui sortent d'un instrument. Les logiciels sont écrits par des êtres humains sous la forme d'un code source, qui est une forme précieuse de connaissance. Il s'agit en fait d'une forme de connaissance très unique, car elle est conçue pour être comprise par des humains et exécutée par des machines. Comme le professeur Abelson du MIT l'a écrit dans un beau livre en 1985, « les programmes doivent être écrits pour que les gens les lisent, et seulement accessoirement pour que les machines les exécutent[1] ».

Que voulait-il dire en disant cela ? Peut-être est-ce plus facile à comprendre si nous pensons à certains exemples particuliers de code source de logiciel. Regardons par exemple un fragment du code source utilisé sur le module d'atterrissage lunaire de la mission Apollo 11, qui nous a permis d'envoyer un homme sur la lune, est assez ancien. Une partie de ce code source est peu lisible, car il s'agit du langage machine pour les tout premiers ordinateurs des années 1960. Cependant, à côté du langage machine, nous trouvons des commentaires écrits en anglais qui décrivent ce que le logiciel est censé faire. C'est un message d'humain à humain. Ce n'est pas seulement un message pour une machine. Plus récemment, si vous regardez les programmes écrits avec un langage de programmation de plus haut niveau, comme le C, vous pouvez trouver de beaux morceaux de logiciels où le langage a évolué au fil du temps. Vous avez un nom pour la variable, un nom pour la fonction, mais là encore, vous avez besoin de commentaires pour comprendre ce qui se passe, bien que parfois, même avec les commentaires, ce ne soit pas si facile.

Len Shustek, président fondateur émérite du Computer History Museum, a joliment déclaré dans son article fondateur de 2006 sur la conservation des logiciels : « L'accès au code source nous donne une vue de l'esprit du concepteur.[2] » Il est très important de le savoir car, une fois encore, c'est l'ingéniosité humaine qui produit tout cela. Ce n'est pas seulement un outil. C'est bien plus que cela. L'histoire du logiciel est assez courte, contrairement à celle de nombreuses autres disciplines. Dans les années 1960, nous avons pu envoyer un homme sur la lune, et nous avons pu le faire, soit dit en passant, grâce à une femme : Margaret Hamilton. Elle a dirigé l'équipe d'ingénieurs qui a développé les 60 000 lignes de code utilisées dans la mission. Ces 60 000 lignes de code étaient alors suffisantes pour envoyer un homme sur la lune et l'en ramener. Aujourd'hui, quelque 50 ans plus tard, un noyau Linux de plus de 20 millions de lignes de code n'est que l'un des nombreux composants des téléphones que nous avons dans nos poches et qui nous permettent d'envoyer un smiley à un ami ou un message à quelqu'un.

Une raison de cette croissance fulgurante est bien sûr le fait que les logiciels transforment le monde dans lequel nous vivons, mais aussi la naissance il y a plus de 30 ans du mouvement des logiciels libres. Cette naissance a conduit à un incroyable effort collaboratif entre des dizaines de millions de développeurs du monde entier, qui ont travaillé ensemble pour construire l'incroyable infrastructure logicielle que nous utilisons tous aujourd'hui. Un vieil adage dit que nous devrions construire sur les épaules des géants, et c'est ce que nous faisons en réutilisant encore et encore de nombreux composants de précédents travaux réalisés par d'autres dans ce même esprit de science ouverte, même si ce mouvement a commencé bien avant que l'expression « science ouverte » ne devienne un sujet d'actualité.

J'insiste particulièrement sur ce point, car on trouve encore parfois des personnes qui pensent qu'un logiciel n'est qu'un ensemble de données, une séquence de zéros et de uns. Ce n'est pas le cas. C'est bien plus que cela. C'est très spécial, très différent. Les projets de logiciels évoluent dans le temps. Certains projets de logiciels peuvent durer des décennies. L'histoire du développement d'un logiciel, c'est-à-dire comment nous l'avons modifié, qui l'a modifié, quoi, quand et comment, est essentielle à sa compréhension. Les logiciels que nous utilisons aujourd'hui présentent par différents aspects une incroyable complexité. Un logiciel peut être complexe parce qu'il est énorme et comporte des millions de lignes de code. Il peut être complexe car même un programme minuscule dépend parfois, pour remplir sa fonction, d'un large éventail d'autres sous-composants et sous-routines, chacune devant parfois à son tour être développée par un grand nombre d'autres personnes. Nous devons garder à l'esprit qu'en réalité, un logiciel que nous utilisons dans le cadre de nos recherches n'est qu'une mince couche par-dessus un incroyable

ensemble de composants logiciels développés par de nombreuses communautés de développeurs dans le monde.

Pour conclure sur ce point, je dirai que les logiciels sont le fruit de l'ingéniosité humaine. Vous ne pouvez pas comparer le code source d'un logiciel à une simple série de chiffres que vous avez obtenus avec l'un de vos instruments. C'est le fruit d'un travail collectif ! Du point de vue juridique, les logiciels sont protégés par le droit d'auteur, contrairement aux données.

Maintenant que nous avons une vue d'ensemble, penchons-nous sur l'importance des logiciels pour la science. On commence à se rendre compte de combien les logiciels sont aujourd'hui indispensables, dans toutes les disciplines. Cela ne concerne pas seulement l'informatique : la plupart des logiciels de recherche ne sont pas développés par des informaticiens. Ils sont développés par des collègues de nombreuses autres disciplines. Aujourd'hui, je pense qu'il est important de faire savoir que lorsque nous parlons de science ouverte, il faut absolument reconnaître qu'il existe au moins trois piliers essentiels : bien sûr, le libre accès aux articles publiés par nos collègues, ainsi que l'accès sans entrave aux données utilisées dans nos expériences, mais cela ne serait pas complet sans un troisième pilier essentiel, qui est le code source des logiciels utilisés pour manipuler, créer et conserver ces données. C'est le pilier logiciel de la science ouverte.

Le logiciel de recherche est un objet aux multiples facettes. Il peut s'agir d'un outil utilisé par quelqu'un pour créer et analyser des données. Il peut s'agir du résultat d'un effort de recherche en tant que preuve d'un résultat ou parce qu'il incarne de nouveaux algorithmes ou des structures de données révolutionnaires. Il peut encore constituer en lui-même un objet de recherche, lorsqu'on cherche à savoir comment un logiciel est construit correctement. Peu importe notre angle d'approche, nous avons besoin d'accéder au code source du logiciel : *l'open source*, que l'on pourrait définir par l'accès ouvert au code source, même si cette pratique est bien plus ancienne que cela, est alors indispensable. Il faut éviter de réinventer la roue, et accélérer la découverte scientifique, et pour cela nous devons conserver l'historique de tous les codes sources construits, afin de favoriser la reproductibilité des résultats de la recherche. Cela facilite également l'acceptation de ces résultats, en permettant l'accès aux outils utilisés pour les obtenir.

Si nous regardons le monde académique, quels types de besoins pouvons-nous identifier autour des logiciels et du code source ? En fonction de votre profil – chercheur, chef d'équipe ou responsable d'un laboratoire, ou encore dirigeant d'un grand organisme de recherche –, vous aurez besoin de disposer d'endroits où archiver et

référencer le logiciel que vous citez dans un article, pour vous assurer que quelqu'un d'autre pourra trouver exactement le même résultat que vous. Vous voudrez bien sûr être crédité de ce que vous avez fait si quelqu'un utilise votre logiciel. Vous pourriez avoir besoin de reproduire le résultat obtenu par un collègue, ou de l'approfondir. Toutes ces pratiques vous sont nécessaires si vous faites de la recherche. Si vous êtes à la tête d'un laboratoire ou d'une équipe, vous devez en outre produire des rapports, savoir sur quels logiciels vous travaillez, maintenir une page web et suivre des contributions logicielles. Si vous êtes un organisme de recherche, vous devez savoir quels logiciels vous utilisez et à quels logiciels vous contribuez. Il est important de mesurer votre transfert de technologie afin d'avoir une idée de votre impact sur la société, car les logiciels construits dans le cadre de la recherche, comme nous le verrons plus tard, ne servent pas uniquement à la recherche. Ils ont parfois un impact direct sur la société. Ces indicateurs sont également nécessaires pour établir une stratégie de financement et pour l'évaluation de la carrière.

Pour satisfaire tous ces nombreux besoins, il y a beaucoup, beaucoup de choses à faire. Je voudrais commencer par ce que l'on pourrait appeler la partie facile. Nous avons bien sûr besoin d'un entrepôt, un endroit où vous pouvez archiver des logiciels en vous assurant que vous pourrez les récupérer plus tard. Vous ne pourrez pas faire cela en utilisant une plateforme d'hébergement de code classique, comme celles que l'on utilise pour développer des logiciels. De telles plateformes ne sont pas des archives : les projets qui y sont stockés vont et viennent, et même les plateformes elles-mêmes vont et viennent, ce ne sont pas des archives.

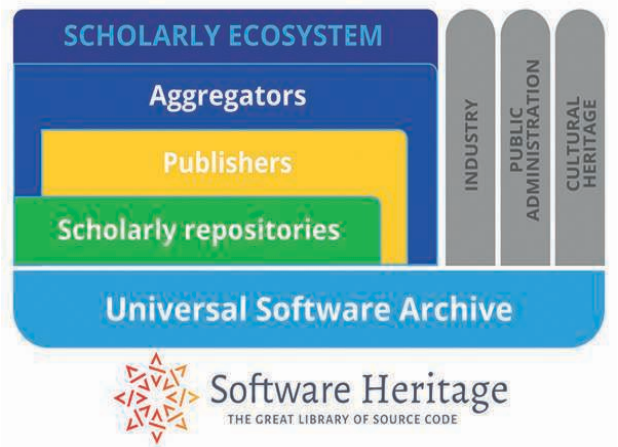
Nous avons besoin d'un moyen de citer exactement, précisément, l'artefact logiciel que nous souhaitons exécuter, pour être sûrs que lorsque nous l'exécuterons à nouveau, nous pourrons reproduire un résultat. Nous devons également fournir une description et des métadonnées appropriées pour faciliter la découverte et la réutilisation de ces artefacts. Enfin, et c'est un point plus délicat parce qu'il est lié à l'évaluation, nous devons trouver la bonne manière de créditer les contributions de ceux qui contribuent aux logiciels de recherche, ce qui n'est pas la même chose que de simplement citer un logiciel.

Si nous voulons faire cela, le point de départ est d'examiner le type d'infrastructures dont nous avons besoin pour soutenir ce type d'usages des logiciels de recherche. Les logiciels sont partout, il y a donc de nombreux écosystèmes impliqués dans ces outils. Il y a un écosystème académique, celui qui nous intéresse aujourd'hui, mais également beaucoup d'autres impliquant, par exemple, l'industrie, l'administration publique, le patrimoine culturel et ainsi de suite. Comme l'a analysé un rapport publié en 2020[3], après six mois de travail d'un groupe de travail européen très large, vous

pouvez repérer les entrepôts académiques, tels que les archives en libre accès, ainsi que les éditeurs et les agrégateurs qui collectent et échangent tous des données. Leur mission, autrefois centrée sur les publications, s'étend désormais aux données, mais qu'en est-il des logiciels ? Si l'on veut gérer correctement des logiciels, il faut travailler avec des archives logicielles universelles, qui relient l'écosystème de la recherche à tous les autres écosystèmes et garantissent un bon archivage et un bon référencement, non seulement pour la fine couche de code source qui aura été développée dans le cadre de travaux de recherche, mais aussi pour tous les autres composants nécessaires à son fonctionnement.

L'initiative Software Heritage que nous avons lancée il y a sept ans entre en jeu ici en archivant tout ce qui est disponible au niveau mondial. Elle fournit une architecture commune à laquelle l'écosystème académique apporte une valeur ajoutée par le biais de la conservation, de la description, de la citation, du crédit, etc. En France, nous travaillons depuis de nombreuses années à la mise en œuvre concrète de cette vision. Il existe un véritable flux de travail à disposition des chercheurs aujourd'hui en France pour archiver automatiquement leurs logiciels, ainsi que les logiciels dont ils ont besoin et qui ne leur appartiennent pas nécessairement, dans l'archive Software Heritage[4]. Ils peuvent également déposer les métadonnées appropriées, qui sont conservées, sur le portail d'accès national HAL[5]. Cela vous permet de générer de belles présentations d'un projet de logiciel, avec les crédits appropriés, avec la référence à l'institut qui l'a financé, une description précise de la manière de le citer et un pointeur qui vous redirige sur cette archive universelle, Software Heritage, qui vous fournit une vue complète de ce logiciel comme de n'importe quel résultat de la recherche, et pas seulement comme d'un tas de zéros et de uns.

Fig. 15 : Infrastructures logicielles de recherche : architecture globale



Source : EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS, « Scholarly Infrastructures for Research Software », Website (Publications Office of the European Union, 7 décembre 2020), http://op.europa.eu/en/publication-detail/-/publication/145fd0f3-3907-11eb-b27b-01aa75ed71a1/language-en.

Cet exemple montre que ces interconnexions peuvent être établies avec succès, et montre également comment le faire correctement. Si vous me le permettez, j'aimerais inviter ici d'autres personnes, organisations et pays à unir leurs forces dans le même type d'initiative. Essayons d'éviter les risques et les erreurs majeurs que nous avons pris et commis dans d'autres domaines, par exemple en tombant dans le piège de la dispersion. Nous ne devons pas construire une multitude d'infrastructures et de silos différents et incompatibles un peu partout. Malheureusement, la tentation est grande pour chacun de construire « sa propre archive », mais le résultat est que vous vous retrouvez avec des artefacts dupliqués au sein de différentes archives, avec des identifiants différents, et vous devrez alors dépenser beaucoup d'argent, de temps et d'efforts pour essayer de construire une fédération après coup au lieu de la concevoir en amont. Nous devrions éviter d'utiliser des plateformes fermées ou à but lucratif que nous ne pouvons pas contrôler, ainsi que d'utiliser l'argent des projets pour financer des opérations, qui sont une question tout à fait différente, dans le secteur de la recherche.

Ce n'est qu'un aperçu des actions qui peuvent être entreprises au niveau infrastructurel. Prenons maintenant un peu de recul, et concentrons-nous maintenant sur les questions politiques plus larges que nous devons également aborder. Si nous voulons vraiment que les logiciels jouent le rôle fondamental qui est le leur dans la science

ouverte, il est important d'avoir des politiques favorables à la diffusion et à la réutilisation des logiciels développés pour faire de la recherche. Nous devons vraiment choisir par défaut l'open source pour les logiciels de recherche. L'open source crée de la valeur : regardez le secteur industriel, qui génère ainsi des milliards d'euros, et vous verrez que ce n'est pas incompatible avec le transfert de technologie. Nous devons simplement adapter à l'open source notre façon traditionnelle de faire du transfert de technologie.

Nous avons également besoin d'un cadre pour l'évaluation et la reconnaissance des chercheurs, car malheureusement, de nombreux pays passent encore du temps à développer de beaux logiciels de haute qualité qui sont nécessaires à la recherche, mais sans que cela compte dans la carrière d'un chercheur ou d'un ingénieur, et cela doit changer. Lorsque nous procéderons à ce type d'évaluation, nous devrons éviter les erreurs que nous avons déjà commises à propos des publications : il importe notamment d'éviter d'avoir uniquement recours à des indicateurs quantitatifs, qui sont encore plus dommageables dans le domaine des logiciels que dans d'autres domaines. Nous devrons également aborder la question de la durabilité de l'open source sur les plans technique, organisationnel et financier.

Il existe toutefois une bonne nouvelle, c'est que la prise de conscience progresse. En 2018, une quarantaine d'experts de toute la planète sont venus à Paris pour travailler sur l'Appel de Paris sur le code source des logiciels[6]. Si vous regardez cet Appel, qui a été publié sur le site de l'Unesco dès 2019, l'un de ses objectifs est de faire reconnaître le développement de logiciels comme une activité de recherche précieuse, et d'en tenir compte dans la carrière des scientifiques s'ils produisent des logiciels de haute qualité. Plus récemment, un rapport sur les infrastructures savantes pour les logiciels de recherche, rédigé par un groupe de travail créé sous l'impulsion de la Commission européenne, a appelé à rendre les logiciels de recherche disponibles en tant que source ouverte, sauf s'il existe de fortes raisons de ne pas le faire.[7]. En outre, dans la recommandation de l'Unesco sur la science ouverte, récemment publiée[8], il est demandé de n'utiliser que des infrastructures à long terme et sans but lucratif pour la science ouverte et de fonctionner à l'échelle communautaire.

La mise en œuvre de ces recommandations de haut niveau a déjà commencé en France. Si vous regardez le 2e Plan national français pour la science ouverte[9], il y a maintenant un chapitre entièrement consacré aux logiciels, qui est au même niveau que la publication et les données. Parmi les nombreuses recommandations figure la création d'une charte pour la politique des logiciels de recherche au niveau national et pour la reconnaissance du développement de logiciels. Ce deuxième point est déjà mis en œuvre, comme vous le verrez lors de la remise des prix du logiciel libre juste

après cette session, et le PNSO inclut a beaucoup d'autres dispositions importantes que je n'ai pas le temps d'approfondir maintenant.

Il y aurait eu tellement d'autres choses à dire, mais je devais en choisir quelques-unes qui me semblent très importantes. Si vous regardez le chemin à parcourir, il est évident que nous devons consacrer plus d'énergie, d'argent et de temps à la construction d'une infrastructure adéquate pour les logiciels de recherche et à la reconnaissance du fait que les logiciels sont un pilier fondamental de la recherche, pas seulement un outil. Les logiciels ont de nombreuses implications, et je pense que l'exposé du professeur Lucke en abordera certaines. Ensuite, nous devons bien sûr nous relier à l'écosystème de la recherche, en connectant le logiciel aux publication et aux données. Le rôle des éditeurs est ici fondamental, mais je leur demande de prendre le temps de considérer que le logiciel est un produit noble de la recherche, pas juste un paquet de données, et qu'il faut lui faut donc des infrastructures et des identifiants spécifiques.

Enfin, nous avons besoin de représentation et de soutien à un niveau institutionnel, de la même manière que cela a pu être fait pour les autres aspects de la science ouverte. Nous avons besoin d'un bureau chargé de la stratégie concernant les logiciels de recherche et les sources ouvertes, et pas seulement en termes de transfert de technologie. Nous devons aider nos collègues à assurer le financement et la gouvernance des logiciels, et ainsi de suite. Enfin, en ce qui concerne les incitations et la reconnaissance dans l'évaluation, il est possible de valoriser les logiciels de recherche de qualité, mais là encore, il faut se méfier des indicateurs quantitatifs. Nous ne voulons pas avoir un indice S(oftware) pour indice logiciel, comme l'indice h dans la publication. Il existe d'autres moyens, comme la cérémonie de remise des prix du logiciel à laquelle vous allez assister aujourd'hui. Nous devons vraiment construire ensemble le pilier logiciel de la science ouverte. Le moment est enfin venu. Comme toujours dans le monde universitaire, le changement prend du temps, mais je crois vraiment qu'ensemble, si nous travaillons de manière cohérente, nous pouvons réellement y arriver.

Pour aller plus loin

« L'étape suivante consisterait à mettre en place une collaboration européenne ou internationale dans le cadre de laquelle les institutions universitaires et les organismes de recherche uniraient leurs efforts et s'appuieraient sur un terrain d'entente en matière d'archivage des logiciels, de référencement des logiciels, de fourniture de métadonnées appropriées, de crédit et de citation, et de reconnaissance du travail des chercheurs pour ce qu'ils font en matière de développement de logiciels. »

« Les prix nationaux décernés à l'occasion des journées OSEC représentent la première fois qu'au plus haut niveau institutionnel, nous mettons en lumière les personnes qui ont passé un nombre incroyable d'heures à construire les logiciels indispensables à la recherche d'aujourd'hui. »

« En dehors du monde universitaire, il y a trop de domaines où les logiciels sont considérés comme de simples outils, et vous n'avez pas beaucoup de respect pour quelque chose qui n'est qu'un outil… Lorsqu'on se rend compte qu'un logiciel est plus qu'un simple outil, on commence à se dire qu'il faut y faire attention. Nous avons donc maintenant besoin de cette impulsion. Nous avons des vice-présidents pour la science ouverte dans les universités, mais nous n'avons pas de vice-président pour l'open source dans le monde universitaire. »

« Il existe de nombreux prix pour les logiciels libres en général, mais c'est la première fois que nous mettons réellement en place un prix pour les logiciels libres dans la recherche à l'échelle d'un ministère. L'objectif est de mettre l'accent sur l'importance des logiciels pour la recherche. C'est une façon de reconnaître les chercheurs qui ont fait un travail incroyable pendant très longtemps sans être reconnus à leur juste valeur dans le milieu universitaire. Le logiciel est très vaste. Ils sont utilisés dans tous les domaines. Bien sûr, nous avons des prix pour les logiciels dans d'autres endroits, mais nous en avons besoin dans le monde universitaire parce que nous devons construire le pilier logiciel de la science ouverte. »

Références

1. Harold Abelson, Gerald Jay Sussman, et Julie Sussman, *Structure and Interpretation of Computer Programs*, The MIT Press, Cambridge, 1985.
2. Len Shustek, « What Should We Collect to Preserve the History of Software? », *IEEE Annals of the History of Computing* 28, nº 4 (octobre 2006): 112-111, https://doi.org/10.1109/MAHC.2006.78.
3. EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS.
4. « Software Heritage », consulté le 24 mars 2022, https://www.softwareheritage.org/.
5. « HAL Science Ouverte », consulté le 16 mars 2022, https://hal.archives-ouvertes.fr/.
6. Software Heritage, « Paris Call: Software Source Code as Heritage for Sustainable Development », 2019, https://unesdoc.unesco.org/ark:/48223/pf0000366715.locale=fr.
7. EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS, « Scholarly Infrastructures for Research Software ».

8. Unesco, « UNESCO Recommendation on Open Science », 2021, https://unesdoc.unesco.org/ark:/48223/pf0000379949.
9. Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation, « Deuxième Plan national pour la science ouverte », *Ouvrir la Science* (blog), 29 juin 2021, https://www.ouvrirlascience.fr/deuxieme-plan-national-pour-la-science-ouverte.