2019-01320 - Doctorant(e) F/H Multi kernels: Evolution at Work

Type de contrat : CDD de la fonction publique
Niveau de diplôme exigé : Bac + 5 ou équivalent
Fonction : Doctorant

A propos du centre ou de la direction fonctionnelle

Le centre de recherche Inria Lille – Nord Europe, créé en 2008, compte 360 personnes dont 305 scientifiques répartis dans 16 équipes de recherche. Reconnu pour son implication forte dans le développement socio-économique sur le territoire du Nord – Pas-de-Calais, le centre de recherche Inria Lille – Nord Europe poursuit une démarche de proximité avec les grandes entreprises et les PME. En favorisant ainsi les synergies entre chercheurs et industriels, Inria participe au transfert de compétences et d'expertises dans les technologies numériques et donne accès aux meilleures recherches européennes et internationales au bénéfice de l'innovation et des entreprises notamment en région.

Contexte et atouts du poste

Inria, institut de recherche dédié au numérique, promeut « l'excellence scientifique au service du transfert technologique et de la société ». Inria emploie 2700 collaborateurs issus des meilleures universités mondiales, qui relèvent les défis des sciences informatiques et mathématiques. Son modèle ouvert et agile lui permet d'explorer des voies originales avec ses partenaires industriels et académiques. Inria répond ainsi efficacement aux enjeux pluridisciplinaires et applicatifs de la transition numérique. Inria est à l'origine de nombreuses innovations créatrices de valeur et d'emplois.

Le centre de recherche Inria Lille – Nord Europe, créé en 2008, compte 360 personnes dont 305 scientifiques répartis dans 16 équipes de recherche. Reconnu pour son implication forte dans le développement socio-économique sur le territoire du Nord – Pas-de-Calais, le centre de recherche Inria Lille – Nord Europe poursuit une démarche de proximité avec les grandes entreprises et les PME. En favorisant ainsi les synergies entre chercheurs et industriels, Inria participe au transfert de compétences et d'expertises dans les technologies numériques et donne accès aux meilleures recherches européennes et internationales au bénéfice de l'innovation et des entreprises notamment en région.

Descriptif de l'équipe

The goal of RMoD is to support ever-running systems. This objective is tackled from two complementary perspectives: reengineering of large systems and constructs for dynamic reflective programming languages.

In the reengineering perspective we propose new analyses to understand and restructure existing large applications (specialized package metrics, adapted visualizations, layer identifications, automated migration) on top of Moose (an open-source reengineering platform) http://www.moosetechnology.org. We work on rule identification, validation. We created Synectique http://synectique.eu a company deploying tools that support software analyses.

In the construct context we are revisiting language concepts such as modules and composition. In addition we are working on new generation of reflective systems. These programming language constructs are experimented on Pharo http://www.pharo.org. We are developing Pharo a dynamically-typed and reflective pure object-oriented language. Pharo is used in several universities worldwide, by research groups and companies. http://consortium.pharo.org is an industrial consortium that supports Pharo.

Keywords:

dynamic languages, language design, reflective programming, security, Smalltalk, Pharo, reengineering, software analysis, program visualization, reverse engineering, meta modeling, moose.

Mission confiée

L'objective de la thèse est de développer une architecture permettant la gestion de plusieurs noyaux de langages de différentes version et leur isolation les uns par rapport aux autres.

Principales activités

Principales activés (5 maximum) :

Activités complémentaires (3 maximum) :

One of the challenge of this PhD is how we can smoothly evolve large ever-running systems that are depending on multiple software sub-projects evolving at different speeds.

We will explore, design and evaluate the idea that ever-running system evolution should be based on the possibility of having multiple versions of such system running side by side in the same runtime infrastructure (virtual machine).

The PhD will focus on the infrastructure to support ever-running software dynamic evolution based on multiple program versions and object migration. This working hypothesis is based on the observation of Gemstone databases which can manage multiple class versions and question their generalisation to ever-running systems.

Informations générales

- **Thème/Domaine :** Programmation distribuée et génie logiciel
  Ingénierie logicielle (BAP E)
- **Ville :** Villeneuve d'Ascq
- **Centre Inria :** CRI Lille - Nord Europe
- **Date de prise de fonction souhaitée :** 2019-09-01
- **Durée de contrat :** 3 ans, 1 mois
- **Date limite pour postuler :** 2019-06-30

Contacts

- **Equipe Inria :** RMOD
- **Directeur de thèse :**
  Ducasse Stéphane /
  stephane.ducasse@inria.fr

A propos d'Inria

Inria, l'institut national de recherche dédié aux sciences du numérique, promeut l'excellence scientifique et le transfert pour avoir le plus grand impact. Il emploie 2400 personnes. Ses 200 équipes-projets agiles, en général communes avec des partenaires académiques, impliquent plus de 3000 scientifiques pour relever les défis des sciences informatiques et mathématiques, souvent à l'interface d'autres disciplines. Inria travaille avec de nombreuses entreprises et a accompagné la création de plus de 160 start-up. L'institut s'efforce ainsi de répondre aux enjeux de la transformation numérique de la science, de la société et de l'économie.

L'essentiel pour réussir

Compétences

- Maîtriser Pharo
- C, Assembleur, concurrence,
- Pratiquer les tests unitaires
- Utilisations Design Patterns (visitor....)

Qualités

- Sens de l'organisation, autonomie, rigueur
- Goût du travail en équipe
- Savoir écouter et communiquer avec des interlocuteurs non techniques ;
- Savoir rédiger des notes / des rapports
- Bonne connaissance de l'anglais

Consignes pour postuler

Sécurité défense :
Ce poste est susceptible d'être affecté dans une zone à régime restrictif (ZRR), telle que définie dans le décret n°2011-1425 relatif à la protection du potentiel scientifique et technique de la nation (PPST). L'autorisation d'accès à une zone est délivrée par le chef d'établissement, après avis ministériel favorable, tel que défini dans l'arrêté du 03 juillet 2012, relatif à la PPST. Un avis ministériel défavorable pour un poste affecté dans une ZRR aurait pour conséquence l'annulation du recrutement.

Politique de recrutement :
Dans le cadre de sa politique diversité, tous les postes Inria sont accessibles aux personnes en situation de handicap.

Attention: Les candidatures doivent être déposées en ligne sur le site Inria. Le traitement des candidatures adressées par d'autres canaux n'est pas garanti.

The PhDfocuses on an important point of the ever-running system life-cycle: the management of their evolution from versions to versions. We will design a new runtime that supports multiple versions of a complex ever-running system running side by side in the same runtime. In a second step, we will improve this infrastructure so that elements can be automatically migrated from using one version to another one.

In addition, programs are not the only entities that need to be evolved dynamically, objects also need this. This raises the question of the possibility for a class to conceptually be able to handle instances of different versions. This notion of multiple versions of classes and their instances raises an important point that can impact the low-level encoding of classes at the object pointer level. We will study the versioning systems of object-oriented databases implementation such as Gemstone from Gemtalk Systems. In particular, we will work on class versioning to allow multiple versions of class to co-exist. We will study and propose explicit migration strategies between class versions. This includes rethinking instance representations so that they embed an explicit version marker.

Related works.

Multiple Versions. Many problems related to evolution are due to the fact that there is only one version of a given component [GB80]. Several efforts have been made to support multiple versions of the same components:

- In Erlang [AVWW96], two different versions of the same software artefact can be active at the same time. When code is loaded in the running system, it retains both the old and new version. Calling conventions define which code is called. This allows for a module to continue executing old code until it is restarted. At most there are two versions active at any time. If a third version is loaded, all processes executing the oldest code are killed. Erlang focuses on providing a robust model for dynamic code loading. It does not try to model change. □
- ChangeBoxes[DGL+07]isoneofthefewattempts[MRH17]tointroducetheideaofmultipleversions running in the same virtual machine at the same time. Changeboxes offer a simple and uniform mech- anism for encapsulating change specifications. They provide a consistent execution scope for running applications, which means that different versions of the same software elements can be simultaneously active within one software system. However, they do not support object migrations and ChangeBoxes semantics should be revisited. We will investigate and define an infrastructure able to run multiple versions of the same application side by side. This infrastructure will support dynamic software evolution and migration at its deep core. Data and Object Evolution. Evolving a running application often requires adapting object structure to new schema. Wide classes have been proposed to migrate instances inside an hierarchy [Ser99]. When objects are stored in databases, this raises the question of data evolution. A number of approaches to modify the conceptual structure of an object database have been developed [Dmi01]: □
- **Schema evolution**: the database has one logical schema to which modifications of class definitions and class hierarchy are applied. Instances are converted (eagerly or lazily, but once and forever) to conform to the latest schema. Examples are O2, GemStone, Objectivity/DB, Versant [Dmi01]. □
- **Class versioning**allows multiple versions of each class to co-exist. Instances can always be represented as if they belong to a specific version of their class, but how this is done (e.g., by creating a separate image of instance for each class version or by keeping one version-specific copy of the instance and dynamically converting it every time it is accessed using a different class version) depends on the concrete system. □
- **Schema versioning**allows several versions of one logical schema to co-exist simultaneously. Similar to the previous approach, instances can be represented in multiple ways. Schema versioning has been explored in Odberg, it was once implemented for the O2 system. In MULTI, we will investigate class versioning and migration of instances between different versions. □

Bibliographie

[AVWW96] Joe Armstrong, Robert Virding, Claes Wikström, and Mike Williams.*Concurrent Programming in Erlang*. Prentice Hall, 1996.

[DGL+ 07] Marcus Denker, Tudor Gîrba, Adrian Lienhard, Oscar Nierstrasz, Lukas Renggli, and Pascal Zumkehr. Encapsulating and exploiting change with Changeboxes. In *Proc. Int. Conference on Dynamic Languages*, pages 25–49. ACM Digital Lib., 2007.

[Dmi01] M. Dmitriev.*Safe Class and Data Evolution in Large and Long-Lived Java Applications*. PhD thesis, Univ. of Glasgow, 2001.

[Ser99] Manuel Serrano. Wide classes. In *Proceedings ECOOP '99*, volume 1628 of *LNCS*, pages 391–415. Springer-Verlag, June 1999.

Compétences

Compétences

- Maîtriser Pharo
- C, Assembleur, concurrence,
- Pratiquer les tests unitaires
- Utilisations Design Patterns (visitor....)

Qualités

- Sens de l'organisation, autonomie, rigueur
- Goût du travail en équipe
- Savoir écouter et communiquer avec des interlocuteurs non techniques ;
- Savoir rédiger des notes / des rapports
- Bonne connaissance de l'anglais

Avantages

- Restauration subventionnée
- Transports publics remboursés partiellement