Offer #2024-08026

PhD Position F/M PhD: Generation of software variants

**Contract type:** Fixed-term contract  
**Level of qualifications required:** Graduate degree or equivalent  
**Other valued qualifications:** PhD  
**Function:** PhD Position

**About the research centre or Inria department**

The Inria Centre at Rennes University is one of Inria's eight centres and has more than thirty research teams. The Inria Centre is a major and recognized player in the field of digital sciences. It is at the heart of a rich R&D and innovation ecosystem: highly innovative PMEs, large industrial groups, competitiveness clusters, research and higher education players, laboratories of excellence, technological research institute, etc.

**Context**

This PhD thesis will be carried out in the DiverSE team [https://www.diverse-team.fr/](https://www.diverse-team.fr/) which is located in Rennes. DiverSE's research is in the area of software engineering.

**Assignment**

Many software systems leverage different mechanisms (feature toggles, compiler flags, configuration files, command-line parameters, etc.) and offer numerous configuration options (or features) that can be combined to generate variants [4]. All these mechanisms aim at augmenting the configurability and features of the system, with positive effects on functionality and performance. The generative nature of LLMs makes them good candidates to produce various software variants with several possible applications: software product lines, self-adaptive system, or simply software systems that want to offer more variants to fit different requirements.

Several research topics are thus of interest in the field of LLM for configurable systems, software product lines, self-adaptive system, or simply software systems that want to offer more variants. A first line of research is that LLM can be used to synthesize software variations based on requirements provided as prompts or high-level feature descriptions. As explored in [1], the idea is to use LLM as compilers capable of synthesizing code variants, corresponding to features, out of prompts written in natural language. First, LLMs can be used to synthesize software variations based on requirements provided as prompts or high-level feature descriptions. As explored, LLMs can act as compilers capable of synthesizing code variants, corresponding to features, out of prompts written in natural language [1]. In [1], we showed how LLMs can assist developers in implementing variability in different programming languages (C, Rust, Java, TikZ, etc.) and mechanisms (conditional compilation, feature toggles, command-line parameters, template, etc.). With "features as prompts", there is hope to raise the level of abstraction, increase automation, and bring more flexibility when synthesizing and exploring software variants. Out of prompts, LLMs can assist developers in implementing variability in different programming languages (C, Rust, Java, TikZ, etc.) and mechanisms (conditional compilation, template-based generator, etc.). The applicability of LLMs for synthesizing code variants seems broad (e.g., we envision to synthesize configuration files in the context of infrastructure as code) but deserves more research. However, there is a major barrier: LLMs are by construction stochastic, non-determinist and highly sensitive to prompt variations -- and so are corresponding implementations of features and variability. A second line is as follows: LLMs can handle high-level requirements and objectives to suggest influential and interpretable configuration options, helping developers make well-informed configuration decisions. Second, LLMs can handle high-level requirements and objectives to suggest influential and interpretable configuration options, helping developers make well-informed configuration decisions. Though statistical learning has been largely employed in this context [3], we believe that LLMs can also bring values when offering recommender and predictive models that estimate software performance based on various configuration settings. These models can communicate to developers or end-users interpretable insights about the complex relationships between configurations and system performance. Hence, LLMs are complementary to statistical learning and symbolic reasoning when handling large variants' space. An interesting perspective is to leverage, in addition to the code, different sources of information (coming from mailing list, documentation, man pages, issues, and discussions) to integrate configuration knowledge. Finally, we will investigate the use of LLMs for automatic feature identification and modeling within software systems. We aim to develop techniques that can identify and represent configurable units (features) effectively [2]. Preliminary experiments suggest that LLMs can either locate features into unmanaged code variants or refactor a configurable system with another set of (meaningful) features.

All of these research directions need further inquiries to propose valid approaches, as variations in prompts or temperature may introduce significant variability-related issues and lead to incorrect
generations. All of these research directions need further inquiries to propose valid approaches. One challenge is that LLMs are sensitive to perturbations, and small variations in prompts may introduce significant variability-related issues and lead to incorrect generated variants. We aim to create specific benchmarks for the task of generating software variants, with the objective of continuously evaluating the robustness of LLMs in handling large variants’ space. It will also guide the development of automated techniques to refine the prompts or provide additional context, thereby enhancing the LLM’s understanding and quality of variants.


Main activities

The PhD candidate will investigate the following research questions:

- How to use LLM to generate software variants?
- What is the most suitable granularity for generating variants?
- What contextual information needs to be built up for effective generation?

Skills

You need to:

- have (or soon receive) a Masters degree in computer science/engineering, informatics, or related fields
- be ok with assisting in teaching and in taking courses where needed
- be ok investing 3+ years as a "research apprentice" (aka PhD student)

Benefits package

- Subsidized meals
- Partial reimbursement of public transport costs
- Possibility of teleworking (90 days per year) and flexible organization of working hours
- Partial payment of insurance costs

Remuneration

Monthly gross salary: 2100€ during the 2 1st years and 2200€ during the 3rd year.

General Information

- Theme/Domain: Distributed programming and Software engineering Software engineering (BAP E)
- Town/city: Rennes
- Inria Center: Centre Inria de l'Université de Rennes
- Starting date: 2024-10-01
- Duration of contract: 3 years
- Deadline to apply: 2024-08-22

Contacts

- Inria Team: DIVERSE
About Inria

Inria is the French national research institute dedicated to digital science and technology. It employs 2,600 people. Its 200 agile project teams, generally run jointly with academic partners, include more than 3,500 scientists and engineers working to meet the challenges of digital technology, often at the interface with other disciplines. The Institute also employs numerous talents in over forty different professions. 900 research support staff contribute to the preparation and development of scientific and entrepreneurial projects that have a worldwide impact.

The keys to success

You need to:

- be really excited about our project
- be persistent (get back up and continue when things don't work out as planned -- true research rarely works out as planned)
- be fearless (e.g., be ok hacking a virtual machine, a compiler, a kernel, or implementing a complex algorithm)
- have a small child's attitude (to want to understand and learn about everything they encounter)
- have an engineer's attitude (not to take the first solution that comes to mind, but to look at the key alternatives)
- have a researcher's attitude (to want to truly understand something, and to not be satisfied with the first best explanation)
- want to look at the simple and obvious before exploring the complicated
- be able to focus (to ignore the many other cool things one could also do)
- derive pleasure from coming up with a logical and clear argument or explanation
- like to read (books, papers, papers, papers)
- like to write (prospectus, proposal, dissertation, and papers)
- like to present (at conferences, or in class)
- like to convince others using sound arguments
- be ok working hard
- under-promise and over-deliver
- be happy staying in Brittany for quite some time
- be ok traveling long distance from time to time (e.g., for conferences)

Warning: you must enter your e-mail address in order to save your application to Inria. Applications must be submitted online on the Inria website. Processing of applications sent from other channels is not guaranteed.

Instruction to apply

Please submit online: your resume, cover letter and letters of recommendation eventually

Defence Security:
This position is likely to be situated in a restricted area (ZRR), as defined in Decree No. 2011-1425 relating to the protection of national scientific and technical potential (PPST). Authorisation to enter an area is granted by the director of the unit, following a favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR would result in the cancellation of the appointment.

Recruitment Policy:
As part of its diversity policy, all Inria positions are accessible to people with disabilities.