ECOLE NATIONALE DE L'AVIATION CIVILE

# Proposition de sujet de thèse – 2022

## Contact

Equipe de recherche : Laboratoire de l'ENAC / Equipe Informatique Interactive
Directeur de thèse : M. Daniel Prun
Contact : daniel.prun@enac.fr

Date de démarrage : rentrée 2022
Financement : bourse de l'école doctorale EDSYS pour 3 ans

## Sujet de thèse

**Titre** : Vérification formelle de systèmes interactifs par approche déductive
**Mots clefs** : Logiciels réactifs, Vérification formelle déductive, Plus faible précondition

L'objectif de cette thèse est de contribuer à la vérification formelle des propriétés graphiques sur les langages réactifs en développant une approche déductive reposant sur le calcul des plus faibles préconditions. Cet objectif est original car d'une part l'approche vise à adapter aux langages réactifs une approche de vérification formelle qui est classique dans le domaine des langages impératifs ; et d'autre part elle cherche à vérifier formellement des propriétés relatives à l'interface homme-machine, particulièrement les propriétés graphiques.

### 1/ Contexte

L'équipe d'Informatique Interactive (LII) du laboratoire de recherche de l'ENAC (Ecole Nationale de l'Aviation Civile) conduit depuis de nombreuses années des recherches en modélisation, analyse et développement de systèmes interactifs. Ces recherches se concrétisent par le développement d'un environnement de modélisation nommé Djnn [1]. Celui-ci permet le développement et l'exécution de logiciels supportant de nombreux types d'interactions homme-machine WIMP et post-WIMP. Il s'articule autour de Smala [2] : un langage dédié à la description de ces logiciels (structuration en composants, mécanismes de contrôle, activation et désactivation de composants, flot de données, gestion des entrées et des sorties…). Compilée puis liée à des librairies de l'environnement Djnn, une application décrite en Smala peut alors être exécutée.
Le langage Smala suit le paradigme des langages réactifs : une application logicielle y est décrite par un ensemble d'instructions qui sont exécutées en fonction d'événements reçus depuis l'environnement (inputs utilisateurs, échéances de timers, réceptions de messages, …), ainsi que de dépendances causales les reliant entre elles (binding pour décrire la relation causale, propagation d'activation, machine à état, etc.). Ainsi, bien qu'il partage certains mécanismes de base présents parmi d'autres langages réactifs comme Lustre, Esterel ou encore RXScala, Smala est caractérisé par son orientation fortement orientée interaction : il produit une représentation perceptible de son état interne par l'intermédiaire de nombreux composants dédiés à la création, l'importation, la manipulation et l'affichage d'objets graphiques et interactifs [3, 4].
Le domaine abordé par cette thèse est celui des systèmes interactifs critiques du transport aérien (systèmes de cockpit d'aéronefs, position de contrôle aérien, interface de pilote de

drones, etc.). Pour pouvoir être utilisés de manière opérationnelle, ces systèmes doivent être certifiés conformément aux directives de la norme DO-178C. La norme complémentaire DO-333 recommande l'utilisation des méthodes formelles pour la vérification des systèmes les plus critiques et encadre leur mise en œuvre.

## 2/ Description

Analyse déductive pour les langages réactifs

Parmi les méthodes de vérification formelle, celle de la vérification déductive consiste à raisonner sur des propriétés du système, par application de techniques de déduction reposant sur des inférences logiques. Floyd [5] et Hoare [6] ont posé les bases de cette approche par l'introduction de la notion de pré et postcondition pour décrire le comportement d'un système. Un triplet {P}S{Q} est utilisé pour exprimer que si un système S est exécuté depuis un état initial où la précondition P est vraie, alors si l'exécution de S se termine dans un état final, la postcondition Q est vraie dans cet état final.

La démarche de vérification consiste alors à partir des préconditions P à démontrer que S conduit aux postconditions Q [7], ou dans l'autre sens, à identifier les préconditions qui satisfont les postconditions attendues. Dijkstra [8] par l'introduction de la plus faible précondition a ouvert la voie à l'automatisation de cette démarche en proposant un moyen de calculer à partir d'une postcondition Q et d'une commande C de S, la précondition minimale assurant Q après l'exécution de C. Il est ainsi possible de générer automatiquement des conditions permettant de respecter un prédicat attendu. De plus, le développement d'outils de type solveurs SAT/SMT ainsi que d'assistants de preuve a contribué au développement de cette approche en permettant de vérifier automatiquement la satisfiabilité de ces conditions, et ce sur des systèmes de taille industrielle.

Cependant, la démarche de vérification basée sur le calcul des plus faibles préconditions concerne principalement les langages impératifs et orientés objets tels que Pascal, Java (projet Loop), C (environnement Frama-C) ou ADA (outil Spark) (un état de l'art complet est proposé par [9]). Le calcul des plus faibles préconditions est défini pour les séquences d'instructions qui sont propres à ces langages (affectation, séquence d'instructions, boucle, instruction conditionnelles). Dans le cadre des systèmes interactifs, les langages de programmation qui sont employés suivent quant à eux le paradigme de la programmation réactive**,** pour lesquels la vérification basée sur le calcul des plus faibles préconditions reste peu adaptée.

**Dans ce contexte, cette thèse se propose de contribuer au développement des techniques d'analyse déductives appliquées aux langages qui suivent le paradigme réactif.**

Etudes des propriétés graphiques

Une autre partie de la problématique concerne les propriétés qui sont objet de la vérification formelle. Historiquement, celle-ci est dédiée à la vérification de la sûreté (e.g. absence d'événement indésirable, bornitude) ainsi que de la vivacité (e.g. retour à un état donné, absence d'interblocage), et adressent les états internes des systèmes. Cependant, l'évolution de ces systèmes et l'apparition des IHM (Interfaces Homme- Machine) modernes font émerger de nouvelles propriétés qui challengent ces méthodes. En particulier, les propriétés graphiques qui portent sur la représentation graphique des données ou de l'état du système (impliquant des notions de couleur, position, forme, visibilité, etc…) restent peu couvertes par les techniques formelles, et leur vérification reposent encore majoritairement sur la mise en place de tests utilisateurs.

**Dans ce contexte, l'objectif est d'appliquer les techniques d'analyse déductives développées dans cette thèse à la vérification formelle des propriétés graphiques des systèmes interactifs.**

## 3/ Démarche proposée

Après un état de l'art approfondi sur les différents concepts mis en jeu dans ces travaux de recherche (vérification par analyse déductive, calcul de plus faibles préconditions, langages réactifs, propriétés graphiques), il s'agira de proposer une sémantique pour les langages réactifs s'appuyant sur la définition des plus faibles préconditions, et de proposer une méthode d'analyse outillée reposant sur cette sémantique. La démarche sera incrémentale en abordant dans un premier temps les instructions basiques des langages réactifs (activation, affectation, …), puis dans un deuxième temps, les instructions plus complexes (machines à état par exemple).
En parallèle, les propriétés graphiques seront étudiées. En particulier, il s'agira de définir un langage d'expression permettant de servir de base à la vérification formelle.
La méthode sera développée et implémentée dans le cadre de l'environnement Djnn/Smala proposé par l'équipe d'informatique interactive, et démontrée sur des systèmes du transport aérien.
Au niveau des objectifs de publication, une communication dans une conférence nationale sera visée à l'issue de la première année (correspondant aux résultats de l'état de l'art ainsi que la présentation de l'approche), qui sera suivie par un objectif de 2 publications de niveau international.

## 4/ Travaux existants – état de l'art

Travaux de l'équipe Informatique Interactive de l'ENAC

Les premiers travaux de l'équipe ont été menés sur la vérification de propriétés graphiques telles que le non-recouvrement d'éléments graphiques, sur des programmes Djnn [10]. Ces vérifications utilisent des méthodes d'analyse statique sur le graphe représentant un programme Djnn et sont ad-hoc (à revalider pour chaque nouveau graphe, et pour chaque propriété).
Une thèse récente dans l'équipe s'est intéressée à la vérification de propriétés graphiques par analyse statique du code Smala [11]. Elle comprend un travail sur l'identification et la formalisation de certaines de ces propriétés puis propose un algorithme d'analyse statique permettant d'identifier les conditions sur les données d'entrées permettant de valider une propriété donnée. Ce travail s'apparente à la une vérification déductive mais ne formalise pas l'approche par la définition de plus faible précondition. Elle constituera donc une base de départ intéressante pour notre travail.
Pour conclure sur les travaux de l'équipe, une thèse est en cours concernant la vérification formelle du compilateur du langage Smala, avec pour objectif de garantir la préservation de la sémantique du programme source à la compilation (cf. [12] pour les premiers résultats). Le but est de proposer une sémantique des différents composants de base de Smala (activation, propagation, …) et de vérifier certaines propriétés à l'exécution (ex : pas d'incohérence lors de la propagation des modifications – absence de « glitch »). L'objectif est de formaliser cette sémantique dans Coq et de prouver l'équivalence sémantique de compositions de composants.

Travaux internationaux

Concernant le calcul des plus faibles préconditions sur les langages réactifs, peu de travaux ont été réalisés. La principale contribution [13] concerne la définition de la « plus faible précondition réactive » (« reactive weakest precondition ») : dans le cadre du langage de spécification abstraite CIRCUS, les auteurs proposent une théorie pour la modélisation et la vérification de systèmes réactifs dans laquelle un calcul de la plus faible précondition pour des système réactifs est développé. Réalisés dans le cadre d'un langage de spécification abstrait, et hors du domaine des système interactifs, ces travaux constituent une référence mais qui reste assez éloignée de l'application pratique que nous souhaitons proposer.
Notons aussi les travaux modélisant les systèmes interactifs par des raffinements successifs [14, 15] dont la preuve repose sur le calcul des plus faibles préconditions. Même si elle met en œuvre des techniques similaires à celles que nous souhaitons étudier, cette approche reste

différente car centrée sur une démarche « descendante » (par raffinements successifs) qui est à l'opposé de ce que nous souhaitons étudier. De plus, ces travaux abordent les propriétés propres aux modèles de tâches et sont peu centrés sur la vérification des propriétés graphiques, qui nous intéresse particulièrement dans cette thèse.

## 5/ Résultats attendus – apports de la thèse

Les résultats attendus concernent plusieurs domaines :
La définition d'une sémantique des langages réactifs reposant sur le calcul des plus faibles préconditions constituera une originalité car celles-ci n'ont été définies et utilisées uniquement dans le cadre des langages impératifs et à objets, et pas dans celui des langages réactifs.
La définition d'un algorithme de vérification reposant sur la sémantique précédemment définie, permettra d'automatiser la vérification déductive des propriétés. Cela conduira à la définition des bases de la programmation par contrat, dans le cadre des langages réactifs : similairement aux contrats de fonctions, nous pourront définir une approche par « contrat d'interaction » où un composant interactif garantira certaines propriétés sous certaines conditions.
La vérification formelle par approche déductive de propriétés spécifiques des systèmes interactifs, et en particulier celles relatives aux aspects graphiques, constituera aussi une originalité puisqu'aucune approche ne va en ce sens.
Enfin, l'adaptation des travaux aux spécificités du framework Smala/Djnn ainsi que le développement d'un outil de vérification déductive reposant sur nos travaux permettra de démontrer la validité de l'approche proposée, et de disposer un outil de vérification pouvant être mis en œuvre sur un ou des cas d'études représentatifs des systèmes interactifs actuels (proposant des interactions WIMP mais aussi post-WIMP) et donc de valider un passage à l'échelle.

## Bibliographie préliminaire


[1] Magnaudet M., Chatty S., Conversy S., Leriche S., Picard C., Prun D.: Djnn/Smala: A Conceptual Framework and a Language for Interaction-Oriented Programming. Proc. ACM Hum.-Comput. Interact. 2, EICS, Article 12 (June 2018).
[2] Smala: http://smala.io
[3] Conversy S., Garcia J., Buisan G., Cousy M., Poirier M., Saporito N., Taurino D., Frau G., and Debattista J.: Vizir: A Domain-Specific Graphical Language for Authoring and Operating Airport Automations. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18) (2018).
[4] Antoine P. and Conversy S.: Volta: the first all-electric conventional helicopter, More Electric Aircraft (2017).
[5] Floyd, R.W.: Assigning meanings to programs. Proc. Symp. Appl. Math 19, 19–31 (1967).
[6] Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM 12(10), 576–580, 583 (1969).
[7] Burstall, R.M.: Program proving as hand simulation with a little induction. In Information Processing 1974, pp. 308–312. Elsevier/North-Holland, Amsterdam (1974).
[8] Dijkstra, E.: A Discipline of Programming. Prentice-Hall, Upper Saddle River (1976).
[9] Hähnle R., Huisman M.: Deductive Software Verification: From Pen-and-Paper Proofs to Industrial Tools. In: Steffen B., Woeginger G. (eds) Computing and Software Science. Lecture Notes in Computer Science, vol 10000. Springer, Cham. https://doi.org/10.1007/978-3-319-91908-9_18 (2019).
[10] Chatty S., Magnaudet M., Prun D.: Verification of properties of interactive components from their executable code. 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2015), Jun 2015, Duisbourg, Germany. ACM, EICS '15 Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp.276-285 (2015).
[11] Béger P. : Vérification formelle des propriétés graphiques des systèmes informatiques interactifs. Interface homme-machine [cs.HC]. INSA Toulouse. Français. tel-02990362v2 (2020).
[12] Marcon C., Nalpon N., Allignol C., Picard C. : Représentation de programmes SMALA grâce à la théorie des bigraphes. AFADL, En ligne, France. ⟨hal-03457039⟩ (Jun 2021).
[13] Foster, S., Cavalcanti, A., Canham, S., Woodcock, J., & Zeyda, F. Unifying Theories of Reactive Design Contracts. Theor. Comput. Sci., 802, 105-140 (2020).

[14] Ait-Ameur Y., Baron M.: Formal and experimental validation approaches in HCI systems design based on a shared event B model. Int. J. Softw. Tools Technol. Transf. 8, 6, 547–563 (November 2006).
[15] Chebieb A., Aït-Ameur Y.: Formal Verification of Plastic User Interfaces Exploiting Domain Ontologies. 9th International Symposium on Theoretical Aspects of Software Engineering (TASE), Nanjing, China. pp.76-86. hal-02450978 (sept. 2015)