

# THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 641  
*Mathématiques et Sciences et Technologies  
de l'Information et de la Communication*  
Spécialité : *Informatique*

Par

**Josselin ENET**

**Protocol-Based Debugging for Domain-Specific Languages**

Thèse présentée et soutenue à Nantes, le 10/12/2024

Unité de recherche : Laboratoire des Sciences du Numérique de Nantes

## Rapporteurs avant soutenance :

Esther GUERRA Full Professor, Universidad Autónoma de Madrid  
Manuel WIMMER Full Professor, Johannes Kepler Universität Linz

## Composition du Jury :

Examineurs :	Esther GUERRA	Full Professor, Universidad Autónoma de Madrid
	Manuel WIMMER	Full Professor, Johannes Kepler Universität Linz
Autre membres :	Olivier BARAIS	Professeur des universités, Université de Rennes
Dir. de thèse :	Gerson SUNYÉ	Maître de conférence avec HDR, Nantes Université
Co-dir. de thèse :	Erwan BOUSSE	Maître de conférence, Nantes Université
Co-dir. de thèse :	Massimo TISI	Professeur des universités, IMT Atlantique

**Titre :** Débogage pour langages dédiés basé sur les protocoles

**Mot clés :** Langages dédiés, ingénierie des langages, outillage, protocoles, débogage

**Résumé :** L'Ingénierie Dirigée par les Modèles considère les *modèles* comme des artefacts centraux, pouvant être utilisés pour de la génération de code, de l'analyse, etc. Ces modèles peuvent être écrits à l'aide de langages dédiés (DSLs), i.e. des langages simples adaptés à des préoccupations précises. En particulier, les *modèles comportementaux* décrivent les comportements possibles du système. Un outil précieux pour le développement de modèles comportementaux est le *débogueur interactif*, qui permet d'observer et de contrôler une exécution. Cependant, le développement d'un débogueur pour DSL est complexe : une large gamme de services doit être intégrée à la fois au DSL et à un environnement de développement (IDE) permettant

d'accéder à ces services. À ce jour, il n'existe pas de débogueur réutilisable capable à la fois de fournir des opérations de débogage pertinentes pour des DSLs hétérogènes, et de s'intégrer facilement dans des IDEs. Dans cette thèse, nous démontrons d'abord par une analyse qualitative que deux protocoles existants (LSP et DAP) ne permettent pas d'intégrer de manière satisfaisante l'outillage d'un DSL dans un IDE. Nous proposons ensuite une nouvelle architecture de débogage pour DSLs, comprenant un débogueur réutilisable paramétré par le domaine. Il repose à la fois sur une extension de DAP pour l'intégration dans des IDEs existants, et sur un nouveau protocole pour la communication avec les implémentations de DSLs.

**Title:** Protocol-Based Debugging for Domain-specific Languages

**Keywords:** Domain-Specific Languages, Language Engineering, Tooling, Protocols, Debugging

**Abstract:** Model-Driven Engineering considers *models* as central artifacts, which can be used for code generation, analysis, etc. Such models can be written using Domain-Specific Languages (DSLs), i.e. small languages tailored for precise concerns. In particular, *behavioral models* express what the possible behaviors of the system under development are. A precious tool for the development of behavioral models is the *interactive debugger*, enabling the observation and control of an execution. Still, developing a debugger for a DSL is complex: a wide range of services must be integrated with both the DSL itself and an Integrated Development Environment (IDE) giving

access to said services. To this day, there exists no reusable debugger able to both provide meaningful domain-specific debugging operations for heterogeneous DSLs, and be easily integrated within IDEs. In this thesis, we first demonstrate through a qualitative analysis that two existing protocols (LSP and DAP) do not allow to satisfyingly integrate DSL tooling within an IDE. We then propose a novel domain-specific debugging architecture featuring a reusable domain-parametric debugger. It relies on both an extension of DAP for the integration within existing IDEs, and a new language protocol for the communication with DSLs implementations.