


People’s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

**Université Ferhat Abbas — Sétif 1**
Faculty of Sciences · Department of Computer Science

---

## Master’s Thesis

---

**Master 2 in Quantum Computing**

# Grover2Q: From UML Models to Executable Qiskit Code

*A Model-Driven Engineering Pipeline for Quantum Software*

Presented by
**Bouchair HADIL**
**Hadj Mokhneche Ahlem**

Supervised by
**Pr. Chaoui Allaoua**
Université Constantine 2 — Abdelhamid Mehri

**Board of Examiners**
President: Pr Toumi Lyazid, Université Ferhat Abbas — Sétif 1
Supervisor: Pr. Chaoui Allaoua, Université Constantine 2
Examiner: Dr Goudjil Lakhdar, Université Ferhat Abbas — Sétif 1

**Academic Year 2025 / 2026**

# Table of Contents

Chapter 1. Introduction....1
1.1 Context and motivation....2
1.2 Problem statement....2
1.3 Objectives....2
1.4 Contributions....3
1.5 Organization of the manuscript....3
Chapter 2. Background....4
2.1 Model-Driven Engineering....5
2.1.1 Models, Metamodels, and Model Transformation....5
2.1.2 Model transformations (M2M and M2T)....7
2.2 Quantum computing foundations....8
2.2.1 Qubits, gates and circuits....8
2.2.2 Grover’s search algorithm....9
2.3 Enabling tools (EMF, ATL, Acceleo, and Qiskit)....10
2.4 Related work....15
2.4.1 Towards model-driven quantum software engineering....15
2.4.2 A quantum software modelling language: Q-UML....15
2.4.3 A UML profile for quantum circuits....16
2.4.4 Model-driven quantum workflows: QuantME....16
2.4.5 Code generation from UML models....16
2.4.6 Positioning of our approach....17
Chapter 3. Contribution....19
3.1 Overview of the approach....20
3.2 Project structure....20
3.3 The metamodels....21
3.3.1 The class-diagram metamodel (SimpleClassDiagram)....21

3.3.2 The activity-diagram metamodel (SimpleActivityDiagram)....22
3.3.3 The Qiskit metamodel....23
3.4 The models....25
3.4.1 The class model....25
3.4.2 The activity model....26
3.5 The model-to-model transformation (ATL)....27
3.5.1 Structure rules....27
3.5.2 Behaviour rules....28
3.5.3 Helpers and lazy rules....29
3.6 The model-to-text transformation (Acceleo / Python)....30
3.6.1 The Acceleo template....30
3.6.2 The portable Python generator....31
3.7 Case study and results....32
Chapter 4. Conclusion and Perspectives....35
4.1 Summary....36
4.2 Limitations....36
4.3 Perspectives and future work....37
Bibliography....39
Résumé....40
Acknowledgements....41
Dedication....42

# Abstract

Model-Driven Engineering (MDE) is a software engineering approach that promotes the use of *models* as first-class artefacts throughout the software development lifecycle, raising the level of abstraction above platform-specific implementation details. The quantum circuit design gets more complex and needs higher levels of abstraction. The development workflows are still largely manual, error-prone and inaccessible to non-experts in quantum mechanics. Although quantum computing platforms such as IBM Qiskit have rapidly matured, there is still a large abstraction gap between high-level system modelling and low-level quantum circuit implementation.

To bridge this abstraction gap, we propose in this Master's thesis an MDE approach that elevates quantum software development to the model level, separating algorithmic intent from hardware-specific implementation. More precisely, we propose a model transformation from UML Models to Executable Qiskit Code. To this end, we define three metamodels — one for a program's structure, one for its behaviour, and one for the Qiskit target. Then, we propose a chain of two automatic transformations. The first one is an *Atlas Transformation Language* (ATL) model-to-model step that translates the abstract UML models into a Qiskit model, and the second is an *Acceleo* (or Python) model-to-text step that turns the obtained model into runnable code. We demonstrate the pipeline on the *two-qubit Grover* search algorithm. Starting from its *class* and *activity* models, the chain produces a working program without a single line of the circuit written by hand, and it recovers the marked state $|11\rangle$ correctly on every run. It is a small but concrete sign that the abstraction and automation of Model-Driven Engineering can carry over to quantum.

**Keywords:** Model-Driven Engineering, quantum computing, Grover's algorithm, Qiskit, ATL, Acceleo, code generation, UML, model transformation, Quantum Software Engineering.

# Résumé

L'Ingénierie Dirigée par les Modèles (IDM) est une approche du génie logiciel qui place les modèles au cœur du cycle de développement, en élevant le niveau d'abstraction au-dessus des détails d'implémentation propres à chaque plateforme. La conception des circuits quantiques devient de plus en plus complexe et exige des niveaux d'abstraction plus élevés. Les processus de développement demeurent cependant largement manuels, sujets aux erreurs et peu accessibles aux non-spécialistes de la mécanique quantique. Bien que des plateformes de calcul quantique telles qu'IBM Qiskit aient rapidement mûri, un important fossé d'abstraction subsiste entre la modélisation de haut niveau et l'implémentation de bas niveau des circuits quantiques.

Pour combler ce fossé, nous proposons dans ce mémoire de Master une approche IDM qui élève le développement logiciel quantique au niveau des modèles, en séparant l'intention algorithmique de l'implémentation liée au matériel. Plus précisément, nous proposons une transformation des modèles UML vers du code Qiskit exécutable. À cette fin, nous définissons trois métamodèles : l'un pour la structure d'un programme, un deuxième pour son comportement et un troisième pour la cible Qiskit. Nous proposons ensuite une chaîne de deux transformations automatiques : la première, une étape modèle-à-modèle écrite en ATL (Atlas Transformation Language), traduit les modèles UML abstraits en un modèle Qiskit ; la seconde, une étape modèle-à-texte réalisée avec Acceleo (ou Python), transforme ce modèle en code exécutable. Nous illustrons la chaîne sur l'algorithme de recherche de Grover à deux qubits : à partir de ses modèles de classes et d'activités, elle produit un programme fonctionnel sans qu'une seule ligne du circuit ne soit écrite à la main, et retrouve correctement l'état marqué $|11\rangle$ à chaque exécution. C'est un signe modeste mais concret que l'abstraction et l'automatisation propres à l'IDM peuvent s'étendre au domaine quantique.

**Mots-clés :** Ingénierie Dirigée par les Modèles, génie logiciel quantique, UML, transformation de modèles, ATL, Acceleo, Qiskit, algorithme de Grover, génération de code.

# الملخص

تُعدّ الهندسة الموجَّهة بالنماذج (MDE) منهجيةً في هندسة البرمجيات تقوم على استخدام النماذج بوصفها عناصر أساسية طوال دورة حياة تطوير البرمجيات، فترفع مستوى التجريد فوق تفاصيل التنفيذ الخاصة بكل منصّة. وقد أصبح تصميم الدارات الكمومية أكثر تعقيدًا ويتطلّب مستويات أعلى من التجريد، في حين تبقى مسارات التطوير يدويةً إلى حدٍّ كبير، وعُرضةً للأخطاء، وصعبةَ المنال على غير المختصّين في الميكانيكا الكمومية. وعلى الرغم من النضج السريع لمنصّات الحوسبة الكمومية مثل IBM Qiskit، لا تزال هناك فجوة تجريد كبيرة بين النمذجة عالية المستوى للأنظمة والتنفيذ منخفض المستوى للدارات الكمومية.

ولسدّ هذه الفجوة، نقترح في مذكّرة الماستر هذه منهجيةً قائمةً على الهندسة الموجَّهة بالنماذج ترفع تطوير البرمجيات الكمومية إلى مستوى النماذج، وتفصل القصد الخوارزمي عن التنفيذ المرتبط بالعتاد. وعلى نحوٍ أدقّ، نقترح تحويلًا للنماذج من نماذج UML إلى شيفرة Qiskit قابلة للتنفيذ. ولهذا الغرض، نُعرّف ثلاثة نماذج فوقية (ميتا-نماذج): أحدها لبنية البرنامج، وثانٍ لسلوكه، وثالثٌ للهدف Qiskit. ثم نقترح سلسلةً من تحويلين آليين: الأول خطوةُ تحويلٍ من نموذج إلى نموذج مكتوبةٌ بلغة ATL تُترجم نماذج UML المجرّدة إلى نموذج Qiskit، والثاني خطوةُ تحويلٍ من نموذج إلى نص تُنجَز بأداة Acceleo (أو بلغة Python) تحوّل النموذج الناتج إلى شيفرة قابلة للتشغيل. ونطبّق هذه السلسلة على خوارزمية البحث لغروفر بكيوبتين: انطلاقًا من نموذجَي الأصناف والنشاط، تُنتج برنامجًا عاملًا دون كتابة سطرٍ واحد من الدارة يدويًا، وتسترجع الحالة المُعلَّمة |11⟩ بشكلٍ صحيح في كل تنفيذ. وهذا دليلٌ متواضع لكنّه ملموس على أنّ التجريد والأتمتة اللذين تتميّز بهما الهندسة الموجَّهة بالنماذج يمكن أن يمتدّا إلى المجال الكمومي.

الكلمات المفتاحية: الهندسة الموجَّهة بالنماذج، الحوسبة الكمومية، خوارزمية غروفر، Qiskit، ATL، Acceleo، توليد الشيفرة، UML، تحويل النماذج، هندسة البرمجيات الكمومية.