

## QCEdit: Modelagem e Geração de Circuitos Quânticos

Anderson B. de Avila, Adriano K. Maron,  
Renata S. Reiser, Maurício L. Pilla

<sup>1</sup>Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPel)  
Caixa Postal 15.064 – 91.501-970 – Pelotas – RS – Brasil

{abdvila, akmaron, pilla, reiser}@inf.ufpel.edu.br

**Resumo.** *Este trabalho apresenta a evolução no desenvolvimento do editor gráfico QCEdit, suas principais características e otimizações. Introduz-se também a modelagem da integração entre o QCEdit e o VPE-qGM, a qual visa explorar a representação gráfica de algoritmos da CQ na forma de Circuitos Quânticos e a simulação distribuída na forma de processos elementares.*

### 1. Introdução

A Computação Quântica (CQ) idealiza o processamento da informação através de um meio físico (atômico/subatômico) que se comporta de acordo com as leis da Mecânica Quântica [Knill and Nielsen 2002]. A unidade básica de informação, o bit quântico (*qubit*), modela um sistema com dois estados básicos, geralmente escritos na notação de Dirac por  $|0\rangle$  e  $|1\rangle$ . Entretanto, diferentemente do *bit*, o *qubit* pode estar em diferentes configurações de superposição dos seus estados básicos, denotadas por  $\alpha|0\rangle + \beta|1\rangle$ , onde  $\alpha, \beta$  são números complexos cujas amplitudes satisfazem a condição de normalidade  $|\alpha|^2 + |\beta|^2 = 1$ . Intuitivamente, pode-se dizer que o *qubit* pode existir nos estados  $|0\rangle$ ,  $|1\rangle$  e também em um estado de superposição, onde  $|\alpha|^2$  e  $|\beta|^2$  representam as probabilidades de ocorrência associadas aos estados  $|0\rangle$  e  $|1\rangle$ , respectivamente [Knill and Nielsen 2002, Nielsen and Chuang 2003].

Devido as características especiais dos *qubits*, a CQ impõe novas fronteiras com relação à velocidade das computações [Hanneke et al. 2009] quando comparadas com o processamento clássico. Exemplificando, o Algoritmo de Shor [Shor 1997] fatora números inteiros em tempo polinomial sobre o número de dígitos; o Algoritmo de Grover [Grover 1996] realiza uma pesquisa eficiente em listas de elementos. Algoritmos quânticos estão sendo propostos, porém exigem plataformas para interpretação e testes, visto a simplicidade do hardware quântico atual. Justifica-se assim ao estudo de simuladores quânticos, introduzindo novas abordagens para CQ.

O ambiente VPE-qGM (*Visual Programming Environment for the qGM Model*) [Maron et al. 2010, Maron et al. 2011b] foi concebido a partir de conceito de processos elementares descritos no modelo qGM [Reiser and Amaral 2010], voltado a modelagem e simulação de algoritmos da CQ sob uma nova concepção. Entretanto, os conceitos do modelo qGM não tem a mesma abrangência do modelo de Circuitos Quânticos, o qual é o paradigma mais difundido para descrição de algoritmos quânticos. Dessa forma, a integração do VPE-qGM ao editor de Circuitos Quânticos QCEdit (*Quantum Circuit Editor*) permite que usuários não familiarizados com o modelo qGM possam desenvolver algoritmos quânticos no modelo de circuitos e exportá-los para a correspondente representação no modelo de processos do VPE-qGM, explorando as possibilidades de simulação distribuídas introduzidas em [Maron et al. 2011a].

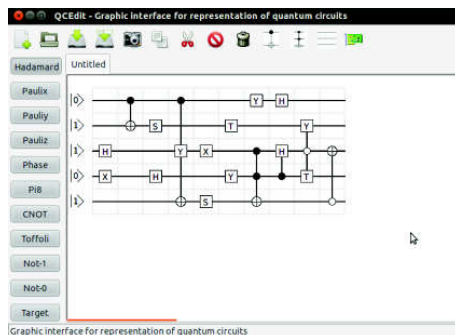
A principal contribuição este trabalho é introduzir a arquitetura do módulo de integração entre *QCEdit* e o *VPE-qGM*, modelada a partir das visão geral obtida com a implementação de funcionalidades deste editor de Circuitos Quânticos.

Na Seção 2 deste artigo, descrevem-se as características do *QCEdit*, incluindo descrição das funções implementadas e a modelagem da integração com o ambiente *VPE-qGM*. Na sequência, a Conclusão com considerações finais e continuidade do projeto.

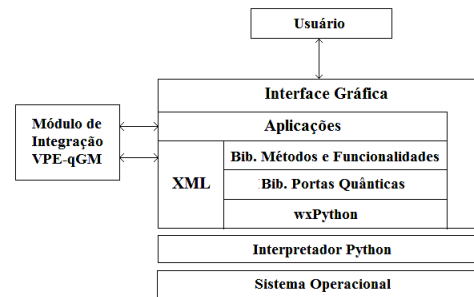
## 2. Editor Gráfico QCEdit

O *QCEdit* é um editor gráfico em desenvolvimento que permite a representação de algoritmos da CQ na forma de circuitos quânticos.

Uma visão da interface gráfica, com exemplificação de circuito quântico contendo portas controladas em sistema de múltiplos *qubits* está ilustrada na Figura 1(a). A visão arquitetural do *QCEdit* está ilustrada na Figura 1(b). Salienta-se a correlação entre três bibliotecas principais: (i) *wxPython*, para suporte a representação gráfica dos circuitos; (ii) Portas Quânticas, para validação e geração de portas quânticas; e (iii) Métodos e Funcionalidades, para aplicação de operadores de edição, incluindo a exportação via arquivos XML para integração com o *VPE-qGM*.



(a)



(b)

Figura 1. Interface e estrutura do editor gráfico QCEdit

### 2.1. Funcionalidades adicionadas ao editor QCEdit

Esta sessão apresenta as extensões para suporte às funcionalidades do *QCEdit*, as quais foram embasadas no estudo de conceitos básico da CQ e da linguagem Python [Ascher and Lutz 2010], em especial sua biblioteca gráfica *wxPython*. Possibilitando a implementação das operações necessárias para reestruturação e atualização das interfaces das ferramentas. As principais funções para modelagem de Circuitos Quânticos são brevemente descritas na sequência:

1. Criação simultânea de circuitos para edição gráfica de algoritmos quânticos e extensão/geração de outros algoritmos;
2. Inserção e deslocamento de portas quânticas pela atuação do mouse;
3. Construção de portas controladas pela implementação de dois operadores: (a) “group”, que agrupa somente as portas selecionadas, eliminando seus controles; (b) “groups”, que além das portas selecionadas, agrupa também seus respectivos controles;

4. Operações de deleção de portas: (a) “*delete gate*”, que apaga somente as portas selecionadas, preservando seus controles e fazendo os ajustes necessários; (b) “*delete gates*”, que além das portas selecionadas, deleta também os controles e portas associadas;
5. Operações de edição: (a) a operação de cópia armazena internamente as portas selecionadas e seus respectivos controles e portas associadas; (b) a operação de recorte é uma operação de cópia seguida da operação “*delete gates*”;
6. Expansão e ajuste automáticos no dimensionamento gráfico dos circuitos, considerando parâmetros como o número de *qubits* e de operadores de cada aplicação;
7. Configuração e alteração de estados iniciais pela disponibilidade de novas janelas de edição;
8. Geração de arquivo XML para salvar o circuito editado, e eventualmente, para novo acesso. Este arquivo XML poderá, posteriormente, carregar o circuito no *VPE-qGM* e viabilizar a simulação da aplicação.

Algumas funcionalidades ainda devem ser adicionadas ao editor, com destaque para a opção de criação, por parte do usuário, de portas quânticas, o que contribui para geração de circuitos mais eficientes e maior agilidade no envio para simulação (sequencial ou paralela) no *VPE-qGM*. A opção criação de macros também viabiliza a geração de portas genéricas, sem necessidade de visualizar as operações envolvidas. Além, é claro, das opções de desfazer e refazer, as quais muito colaboram na construção e edição dos circuitos.

## 2.2. Módulo de integração dos ambientes QCEdit e VPE-qGM

Esta seção apresnta a modelagem do módulo de integração entre os ambientes, permitindo a exportação de algoritmos quânticos de uma ferramenta para outra via a aplicação corrente ou exportação/importação do correspondente arquivo XML. Uma representação deste modulo pode ser vista na Figura 2, considerando a implementação do módulo *qGM – Converter* [Maron et al. 2011b].

A aplicação do método *XMLParser* converte a entrada de dados, no formato XML, para a estrutura matricial considerada pelo método *QC.to.qGM*, cuja execução chama os construtores do *qPE* de acordo com a ordenação dos operadores em *M*.

**Figura 2. Visão arquitetural da integração entre os ambientes**

### 3. Conclusão

A abordagem seguida neste trabalho caracteriza os esforços iniciais para expandir as capacidades de modelagem oferecidas pelo ambiente *VPE-qGM*. O trabalho descreve a extensão de funcionalidades do *QCEdit*, viabilizando a descrição dos algoritmos em uma representação de mais alto nível. Usufruindo de operações que facilitam a descrição de algoritmos quânticos, consolida-se uma solução voltada para a simplificação do processo de edição, tornando mais intuitiva a análise, interpretação e compreensão de algoritmos. Assim, tem-se a facilidade de criação do editor *QCEdit* e as possibilidades de simulação distribuída do *VPE-qGM*, permitindo a criação de algoritmos da *CQ* segundo o modelo de Circuitos Quânticos. Viabiliza-se então o acesso ao ambiente de simulação *VPE-qGM* independente de um estudo sobre a utilização de processos elementares para representação de algoritmos quânticos. Na continuidade, considera-se a implementação e consolidação da integração entre *VPE-qGM* e o *QCEdit*.

### Referências

- Ascher, D. and Lutz, M. (2010). *Aprendendo Python*. Bookman.
- Grover, L. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219. Disponível por WWW em <http://doi.acm.org/10.1145/237814.237866> (dec.2011).
- Hanneke, D., Home, J., J.D., J., Amini, J., Leibfried, D., and D.J., W. (2009). Realizations of a programmable two-qubit quantum processor. *Nature Physics*, 6:13–16.
- Knill, E. H. and Nielsen, M. A. (2002). Theory of quantum computation. <http://www.citebase.org/abstract?id=oai:arXiv.org:quant-ph/0010057>.
- Maron, A., Pinheiro, A., Reiser, R., and Pilla, M. (2010). Modelagem e simulação quântica no ambiente vpe-qgm. In *Workshop Escola de Computação e Informação Quântica*, pages 121–130.
- Maron, A., Pinheiro, A., Reiser, R., and Pilla, M. (2011a). Consolidando uma infraestrutura para simulação quântica distribuída. In *Anais da ERAD 2011*, pages 213–216. SBC/Instituto de Informática UFRGS.
- Maron, A., Ávila, A., Reiser, R., and Pilla, M. (2011b). Introduzindo uma nova abordagem para simulação quântica com baixa complexidade espacial. In *Anais do DINCON 2011*, pages 1–6. SBMAC.
- Nielsen, M. A. and Chuang, I. L. (2003). *Computação Quântica e Informação Quântica*. Bookman.
- Reiser, R. and Amaral, R. (2010). The quantum states space in the qgm model. In *Anais/III Workshop Escola de Computação e Informação Quântica*, pages 92–101, Petrópolis/RJ. Editora do LNCC.
- Shor, P. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*.