

Análise de Bibliotecas JavaScript e Implementação Inicial de um Editor de Processos Quânticos Elementares como Serviço

Gustavo dos Santos, Anderson Ávila, Renata Reiser

¹Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas (UFPEL) – Pelotas – RS – Brazil

{gfdsantos, abdavila, reiser}@inf.ufpel.edu.br

Resumo. Neste trabalho é apresentado um panorama sobre o estado atual do editor de processos elementares presente no ambiente VPE-qGM. É apresentado o desenvolvimento de uma solução multiplataforma em orientada a serviço. É feita uma breve discussão sobre as bibliotecas existentes para diagramação on-line e, é mostrado uma visão sobre uma implementação alternativa do qPE.

1. Introdução

O acesso a computadores quânticos é limitado pelo alto custo de aquisição e manutenção. Por outro lado, a evolução deste ramo da ciência não pode ser deixado de lado. Faz-se necessário o uso de simuladores quânticos e, neste sentido, este trabalho visa mostrar os primeiros passos na construção de um editor de processos orientado a serviço.

O ambiente VPE-qGM tem como objetivo a modelagem e simulação de circuitos quânticos baseado no modelo qGM [Vieira 2015]. Dos principais componentes do ambiente VPE-qGM (Virtual Programming Environment for the Quantum Geometric Machine), existe o qPE - *quantum Process Editor*. O objetivo do qPE é ser um módulo do VPE-qGM que proporciona a diagramação de circuitos quânticos. Um circuito quântico é exportado para um arquivo formatado do tipo XML. O módulo qPE é capaz de carregar para a memória um circuito quântico salvo em um arquivo XML.

O módulo qPE possui diversas portas quânticas e têm a habilidade de manipular relações entre qubits e as portas quânticas, formando unidades de computação quântica. É escrito em Python 2 e faz uso extensivo da biblioteca wxPython para construções gráficas. Com as atualizações desta biblioteca usada como base, a compatibilidade do módulo qPE com as versões recentes do wxPython foi se perdendo. A biblioteca wxPython proporciona o desenvolvimento de interfaces gráficas a nível de sistema operacional, usa recursos gráficos deste sistema operacional para fazer a montagem da interface de usuário. Por conta destes fatores limitantes, para implementação do editor de processos, é necessária uma implementação usando uma biblioteca para desenvolvimento web.

Este trabalho propõe uma nova implementação do módulo qPE, desta vez como parte de um serviço acessível via internet. Houve um período de análise das bibliotecas e tecnologias web disponíveis atualmente para implementação de uma interface de diagramação online. Após ser decidido a biblioteca a ser usada ao longo do desenvolvimento, foi construído um protótipo. As etapas de análise e desenvolvimento dos protótipos é discutido ao longo deste trabalho, bem como os próximos passos deste projeto.

2. Motivação

O simulador quântico VPE-qGM possui um módulo gráfico para editar processos elementares. Este módulo é escrito em Python 2, e usa a biblioteca wxPython para as construções gráficas. Contém uma interface completa e possibilita a diagramação eficiente de processos quânticos. Visando o crescente uso da computação em nuvem, a motivação deste trabalho é dar os primeiros passos na implementação de um editor de processos online hospedado como um serviço, no qual pode vir a ser uma alternativa ao qPE.

A disponibilização de um software como serviço via internet viabiliza o uso de aplicações independente de plataforma. Foi optado pelo desenvolvimento do editor em tecnologias web, que pode tanto estar disponível em um serviço na internet, como também pode ser empacotado em um executável para várias plataformas. Um exemplo de ferramenta que empacota um aplicativo web em aplicativo executável em sistemas operacionais é o Electron [Electron].

O objetivo com o desenvolvimento de um novo editor de processos é disponibilizá-lo para o acesso de usuários em geral. Para tal é necessário que a compatibilidade visual seja mantida entre o módulo qPE atual e a implementação web do qPE, ou seja, é necessário uma análise prévia de bibliotecas disponíveis atualmente que permitem o desenvolvimento completo da interface do editor de processos e que esta seja completamente funcional no navegador instalado no sistema operacional hospedeiro do usuário. A compatibilidade de funcionalidades é outro fator limitante para a escolha da biblioteca adequada ao projeto. A biblioteca deve proporcionar características como salvar o estado atual do diagrama de processos elementares, carregar o diagrama de processos a partir de um arquivo formatado corretamente ou então permitir agrupamento de processos sem perda de informações, tal que habilite a possibilidade de desfazer esta operação.

Foram analisadas diversas bibliotecas a fim de descobrir o quanto poderiam suprir os requisitos do projeto. Algumas destas bibliotecas se mostraram ser eficientes em termos de quantidade de código necessário para realizar uma determinada função e em termos das funções que conseguem executar. Uma explicação extensiva sobre todas as bibliotecas analisadas não contempla o objetivo deste trabalho. Das bibliotecas analisadas, foram escolhidas três que combinaram melhor suas funcionalidades com os requisitos necessários para o projeto a serem mencionadas nesta sessão. A seguir é feito um breve comentário sobre cada uma das bibliotecas de destaque.

- **GoJS.** Biblioteca JavaScript de diagramação em aplicativos web. Possui um rico sistema de animações e objetos visuais. Não usa qualquer outra biblioteca de JavaScript e é altamente otimizado [GoJS]. Possui uma vasta lista de exemplos, dos mais variados tipos. Esta biblioteca é licenciada com base em uma ativação de licença paga, o que a torna fora do escopo do projeto.
- **JsPlumb.** Biblioteca JavaScript para manipulação de HTML. Possui uma versão comunitária e um kit de ferramentas que pode ser adquirido com uma licença. A versão comunitária é de código aberto e de livre uso [jsPlumb].
- **MxGraph.** [JGraph] Biblioteca de diagramação web que executa por completo no cliente. É baseado em arquivos do tipo SVG e páginas HTML, não excluindo a possibilidade de uso desta biblioteca com outras tecnologias web para construção de *web apps* dinâmicos, como React.js ou Angular.js. É de código aberto e de livre uso.

A biblioteca mxGraph atendeu a todos os requisitos, no qual mostrou-se ser eficiente ao implementar as primeiras funcionalidades. Foi definido como a biblioteca base para a construção do editor de processos proposto neste trabalho.

3. Metodologia

O desenvolvimento do protótipo se deu usando um *framework* JavaScript chamado mxGraph. Como já discutido neste trabalho, foi buscado uma biblioteca capaz de efetuar desenhos gráficos a partir de polígonos, com capacidade de obter objetos selecionados, agrupamento e capacidades gráficas para construção de interfaces de usuários avançadas em navegadores. MxGraph se mostrou competente e cumpriu com os requisitos nos testes preliminares.

Construir um editor de processos online exige que o navegador do usuário consiga manipular os objetos gráficos sem que o desempenho seja demasiado alterado. MxGraph mostrou que não exige que o navegador do usuário utilize muitos recursos de hardware do sistema operacional hospedeiro.

A construção básica implementada neste trabalho é baseada em um grafo principal, que faz a interligação entre os itens. Cada item é do tipo *Vertex* e possui um estilo, não excluindo a opção da existência de múltiplos itens de mesmo estilo. No protótipo atual existem dois estilos criados, para melhor visualização de diferentes itens.

Das operações que mxGraph habilita o usuário executar sobre o grafo, neste projeto utilizamos apenas operações de adição e remoção. Cada vez que o usuário clica em um item na barra lateral e o solta na área de trabalho, um item é adicionado ao grafo. Operações de agrupamento e desagrupamento também executam apenas as operações previamente mencionadas, onde uma operação de agrupamento remove todos os itens selecionados no grafo e os substitui por um novo item, no qual simboliza um grupo. Um grupo contém uma coleção de itens e suas informações de identificação, como um número de identificação. Um número de identificação de item no grafo é um contador interno do próprio grafo. Cada item adicionado ao grafo causa um incremento deste contador. O contador não decrementa, mesmo após remoção de itens do grafo. Um protótipo do editor pode ser visto na Figura 1.

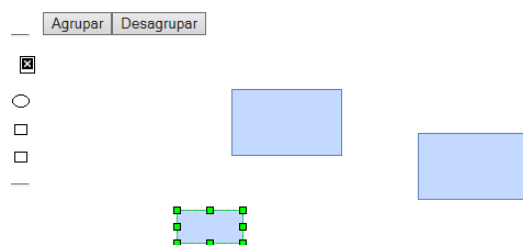


Figura 1. Protótipo do novo editor de processos elementares

O desenvolvimento do editor pode ser acompanhado através do repositório hospedado no GitHub[Santos]. É de código aberto e de livre uso, logo a colaboração da comunidade neste projeto é livre.

Um usuário que terminar o seu trabalho de edição, pode exportar seu diagrama de processos para um arquivo formatado do tipo XML. Este arquivo serve como entrada para o simulador VPE-qGM.

4. Conclusão

O desenvolvimento do editor vem se dando de forma rápida e novas funcionalidades são constantemente adicionadas ao projeto. É necessário visualizar o ciclo de desenvolvimento ao longo do tempo e, os próximos passos do grupo engloba a construção das portas quânticas com base em polígonos, que por sua vez são construídos dinamicamente no navegador do usuário. Busca-se também implementação da funcionalidade de salvar o trabalho corrente do usuário e também restaurar diagramas do usuário. Esta funcionalidade será implementada seguindo o padrão de arquivo que o simulador VPE-qGM toma como entrada. Este componente deve também receber como entrada um arquivo formatado do tipo XML e a partir deste construir o circuito de forma gráfica, mantendo todas as construções do usuário no local onde elas estavam anteriormente.

Busca-se a hospedagem do editor de processos como um serviço na internet. Uma análise das formas disponíveis no mercado para hospedagem de um aplicativo web faz parte dos próximos objetivos buscado pelo grupo.

5. Referências

Referências

- Electron. Biblioteca open source desenvolvida pelo o github para o desenvolvimento de aplicações desktop multi-plataforma com html, css e javascript.
- GoJS. Interactive javascript diagrams in html.
- JGraph, L. mxgraph - an open source javascript diagramming component.
- jsPlumb, I. Jsplumb.
- Santos, G. Implementação web do vpe-qgm qpe.
- Vieira, J. (2015). Análise e simulação dos conectivos fuzzy xor e classes derivadas via computação quântica.