

# Análise Lumínica Virtual de Elementos Construídos por Meio de Programação: Exemplo de Aplicação em Software do Tipo BIM

Arq. Momchil Stoyanov  
stoyanov@hotmail.com

**Abstract.** *“Daylighting is an important part of designing sustainably. Daylighting is the use of natural light for primary interior illumination. This reduces our need for artificial light within the space, thus reducing internal heat gain and energy use. Direct sunlight, once it enters the building, is not only light but heat, and that additional heat will need to be taken into account in your energy analysis.” While Autodesk Revit Architecture 2010 (ARA) itself cannot perform the actual analysis, there are some ways to do that. This paper focuses on the study of parametric modeling using a BIM tool for daylighting analysis. This paper presents the first part of the building method of LUME, a plug-in made with C# programming language in Microsoft Visual Studio 2010 and ARA Software Developer Kit (SDK) package. The script accepts as its input a standard three dimensional model of building opening and his position on space.*

**Keywords:** Script language; BIM; Revit Architecture; Energy analysis; Daylighting; Parametric design process

## 1. Introdução

Com o desenvolvimento da Tecnologia da Informação (TI), há um número cada vez maior de ferramentas que podem ser utilizadas no processo de projeto e, também, nas etapas posteriores de cálculo, produção e construção. As áreas de Arquitetura, Engenharia e Construção (AEC) podem se beneficiar de uma série de recursos que vão muito além do uso de computadores para desenhar (Computer Aided Drafting), evoluindo para Computer Aided Design (Projeto Assistido por Computador), Computer Aided Engineering (Engenharia Assistida por Computador) e Computer Aided Manufacturing (Manufatura Assistida por Computador) - CAD, CAE, CAM.

O modelo paramétrico da construção virtual se insere no conceito de CAD e é um tipo de arquivo digital gerado por software do tipo BIM (Building Information Modeling, traduzido em português comumente como Modelagem da Informação da Construção). Programas computacionais do tipo BIM fazem parte de uma geração de ferramentas digitais “inteligentes” orientadas ao objeto que possibilitam ao usuário gerenciar num ambiente virtual informações de projeto relativas à diversas etapas da construção. “BIM é mais do que um modelo para visualização do espaço projetado, é um modelo digital composto por um banco de dados que permite agregar informações para diversas finalidades, além de aumento de produtividade e racionalização do processo” (CRESPO, C.; RUSCHEL, R. C., 2007).

O presente projeto pretende desenvolver uma aplicação ou programa de computador denominado LUME executável em software do tipo BIM. Mais especificamente, por meio da utilização de linguagem de programação e do conjunto de ferramentas de programação incorporadas no software Autodesk Revit Architecture (ARA) é elaborado um aplicativo simples que analisa parâmetros de insolação de aberturas em modelos digitais pré-existentes. O objetivo principal é introduzir parâmetros especificados pelo usuário num software BIM padrão para efetuar testes de desempenho/ performance durante o processo de projeto.

## 2. O aplicativo LUME dentro do Autodesk Revit Architecture

A metodologia proposta inclui três etapas básicas: geração de elementos vazados (paredes com aberturas de diferentes padrões, blocos vazados, etc.) utilizando as ferramentas padrão do ARA; elaboração do aplicativo externo por meio da linguagem de programação C# incluindo banco de dados sobre parâmetros físicos básicos relativos à insolação; teste dos elementos vazados; geração de resultados do teste (ganho de calor por

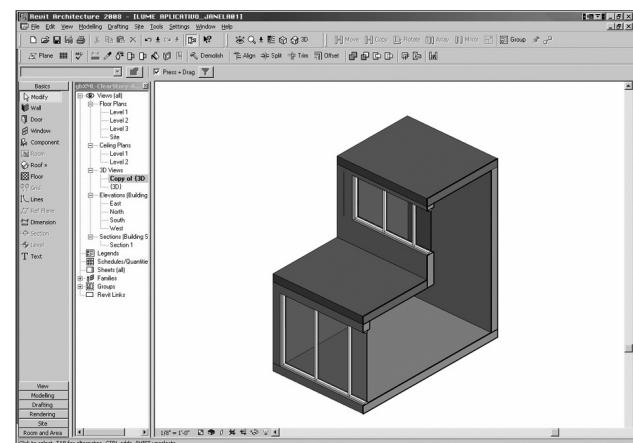


Figura 01 – interface do programa Autodesk Revit Architecture 2008 com um modelo virtual contendo aberturas numa das faces

metro quadrado, gráfico do ganho de calor no decorrer do dia, entre outros). No processo de teste, além do aplicativo externo proposto, são usadas ferramentas padrão do ARA como, por exemplo, orientação solar e simulação de insolação.

O presente artigo engloba a primeira parte do projeto, ou seja, a elaboração da funcionalidade de LUME no que se refere à coleta de dados da abertura ou janela previamente criada no Revit.

Com o aperfeiçoamento do ARA, novas ferramentas de projeto sustentável foram incorporadas no software. Apesar disso, o programa não faz análises lumínicas e sim, possibilita configurar o modelo virtual de forma que seja possível efetuar posteriormente a análise em si, por meio do uso de aplicativos externos. Há certos caminhos que podem ser seguidos para atingir o objetivo de efetuar análise mais sistemática.

Primeiro, o ARA pode exportar o modelo para programas externos como o Autodesk 3dsMax, o Autodesk Ecotect. Segundo, podem ser utilizados plug-ins como o oferecido pela Integrated Environmental Solutions (IES) que efetuam a análise mais aprofundada, inclusive assegurando conformidade com os padrões do Leadership in Energy and Environmental Design (LEED). Mas há, também, outra possibilidade interessante: coletar as informações básicas de insolação do Revit por meio de um aplicativo interno que consegue “ler” ou extrair informações geradas pelas ferramentas padrão do ARA e, efetuar sua análise posteriormente.

<sup>1</sup> DEMCHEK, G., DZAMBAZOVA, T., KRYGIEL, E. Mastering Revit Architecture 2010.

# SIGraDi 2009 sp

O esquema de elaboração de comandos personalizados no ARA ainda é limitado no que se refere à praticidade de uso e produtividade, comparado às ferramentas que programas como o AutoCAD oferecem. No entanto, o aprimoramento é rápido e as vantagens do sistema BIM são grandes. O esquema de programação não ocorre dentro do ARA. É necessário utilizar aplicativos externos. No caso, foi utilizado o Microsoft Visual Studio 2010. O ARA aceita qualquer aplicativo ou comando elaborado por meio do uso da plataforma Microsoft .NET, sendo que no caso foi utilizada a linguagem C#. O comando externo é elaborado nesse programa e exportado para o ARA, sendo que a conexão entre os dois programas é feita por meio da modificação do arquivo de inicialização do ARA (Revit.ini) com código do tipo:

```
[ExternalCommands]
```

```
ECCount=1
```

```
ECName1=LUME
```

```
ECClassName1=Revit.SDK.Lume.CS.Command
```

```
ECAssembly1=Lume.dll
```

```
ECDescription1=Displays LUME application for daylight  
analysis using an Autodesk Revit API command
```

No Revit, caso o CE não apresente erros, surge um sub-menu na barra de ferramentas denominado External Tools que carrega e executa o comando. Portanto, toda a interface e funcionalidade desse comando é elaborada fora do ARA. O processo permite utilizar boa parte dos comandos e funcionalidades do ARA e extrair diversas informações durante o processo de modelagem.

O presente LUME extrai as seguintes informações ou parâmetros do ARA:

Posição espacial do elemento que “recebe” a insolação, a posição dos pontos que formam a sombra resultante disso, a orientação solar, as informações do sol (posição, data e hora do ano). Com isso, é gerada uma base de dados simples. Posteriormente, num processo externo, as informações podem ser organizadas em planilha e transformadas em gráficos com diversas finalidades como, por exemplo, intensidade luminosa, ganho de calor para a amostra em análise (abertura ou janela do modelo virtual).



Figura 02 – interface do programa LUME contendo um formulário (Windows.form) padrão do Windows com um modelo virtual contendo aberturas numa das faces

Essa primeira parte do aplicativo LUME extrai algumas informações básicas do ARA relativas a abertura do modelo ou janela pela qual o ambiente interno recebe luz solar, posição X, Y, Z do perímetro, etc. O aplicativo é um tipo arquivo de classe (Class library) que opera a pasta raiz do ARA fornecida no guia de desenvolvimento da Autodesk (SDK). A principal parte do comando solicita informações sobre a abertura ou aberturas existentes no modelo por meio do seguinte trecho de código em C#, onde parte das variáveis declaradas são identificadas (nome do elemento ou objeto “janela”, elemento receptor da janela ou “parede”):

```
/// </summary>  
public class OpeningProperty  
{  
    private string m_name = "Opening"; // nome da abertura  
    private string m_elementId = ""; // Element Id da abertura  
    private string m_hostElementId = ""; // Element Id do receptor da abertura  
    private string m_hostName = "Null"; // nome do receptor  
    private bool m_isShaft; // tipo de abertura
```

```
/// <summary>
```

Outro conjunto de variáveis declaradas complementam a base de dados com informações relativas à posição da janela no espaço:

```
/// <summary>
```

```
/// seleciona quatro pontos de canto de um retângulo no mesmo  
plano
```

```
/// </summary>
```

```
/// coordenadas do retângulo</param>
```

```
private XYZArray GetPoints(XYZArray boundRect)
```

```
{
```

```
    XYZArray points = new XYZArray();
```

```
    XYZ p1 = boundRect.get_Item(0);
```

```
    points.Append(ref p1);
```

```
    XYZ p2 = new XYZ();
```

```
    p2.X = boundRect.get_Item(0).X;
```

```
    p2.Y = boundRect.get_Item(0).Y;
```

```
    p2.Z = boundRect.get_Item(1).Z;
```

```
    points.Append(ref p2);
```

```
    XYZ p3 = boundRect.get_Item(1);
```

```
    points.Append(ref p3);
```

```
    XYZ p4 = new XYZ();
```

```
    p4.X = boundRect.get_Item(1).X;
```

```
    p4.Y = boundRect.get_Item(1).Y;
```

```
    p4.Z = boundRect.get_Item(0).Z;
```

```
    points.Append(ref p4);
```

```
//make rectangle close
```

```
XYZ p5 = boundRect.get_Item(0);
```

```
points.Append(ref p5);
```

```
return points;
```

```
}
```

Essa primeira parte do LUME coleta um conjunto de informações que servirão como base nas etapas posteriores da própria análise luminosa como, por exemplo, dimensões da janela, posição no espaço, etc. O desafio a seguir será como adicionar características “inteligentes” ao programa com o objetivo de coletar informações mais completas e detalhadas a partir do modelo virtual para efetuar a análise luminosa final.

### 3. Conclusão

O presente artigo busca ilustrar de forma simples o processo de programação do ARA e é uma tentativa de explorar e aprimorar a capacidade do software de testar elementos construtivos virtualmente numa busca por eficiência na construção. Dessa forma, procura-se ampliar o horizonte do usuário de softwares padrão, mostrar parte do potencial das ferramentas BIM, exemplificar e discutir uma das questões-chave colocadas por Branko Kolarevic no XI congresso da Sociedade Ibero Americana de Gráfica Digital (apresentação intitulada Generative + Performative, México, outubro de 2007) – a incorporação de conceitos da física, e não apenas da geometria, em programas de modelagem tridimensional.

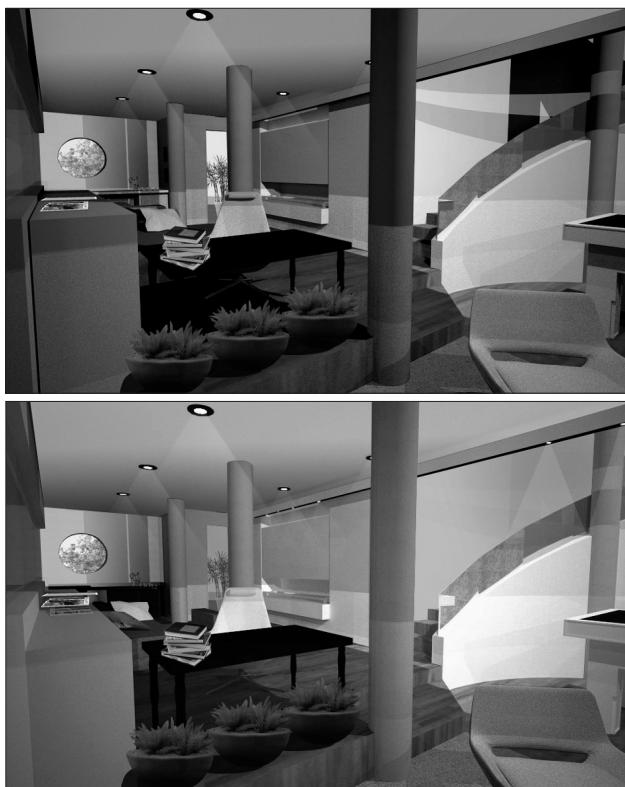


Figura 03 – ambiente com luz artificial e luz natural (com relação à Figura 04 varia apenas a intensidade da luz natural).

O uso correto de luz solar é fator importante de um projeto arquitetônico sustentável. O uso dessa fonte primária de luz nas edificações pode reduzir a necessidade de iluminação artificial diminuindo os ganhos de calor e economizando energia elétrica. A insolação direta é uma fonte não apenas de luz, mas, também, de calor. É portanto, fundamental, no decorrer do projeto de arquitetura elaborar análises lumínicas dos ambientes, com o objetivo, entre outros, de assegurar iluminação adequada para as atividades que serão ali exercidas, sem prejudicar o conforto térmico.

Um bom projeto lumínico leva em consideração uma série de fatores como, por exemplo, orientação do prédio, características das fachadas, volumetria. Compreender o efeito da luz solar sobre o edifício é primordial para a elaboração de projetos arquitetônicos sustentáveis.

O uso de programas do tipo BIM para auxílio de projeto é interessante porque os mesmos conseguem relacionar informações de determinado banco de dados com o modelo virtual em tempo real e, dessa forma, otimizar o processo de simulação ou teste do edifício. Assim, o potencial para exploração e aprimoramento de análises lumínicas em ambiente BIM é grande.

### Referências

- KOLAREVICH, B. *Architecture in the Digital Age: design and manufacturing*. London, Spon Press, 2003.  
 MITCHELL W. J. *The Logic of Architecture*. Cambridge, MIT Press, 1990.  
 TERZIDIS, Kostas. *Algorithmic Architecture*. Elsevier, Oxford, 2006.  
 Internet  
 Wikipedia - <http://pt.wikipedia.org/>  
 TAMMIK, Jeremy. *The Building Coder* - <http://thebuildingcoder.typepad.com/>  
 Microsoft Developer Network MSDN - <http://msdn.microsoft.com/pt-br/default.aspx>  
 Autodesk Developer Center – [www.autodesk.com](http://www.autodesk.com)

### Referencias Figuras

Figura 03 e Figura 04 – maquete eletrônica com autoria de Arq. Momchil Stoyanov e projeto arquitetônico de Arq. Sabrina Baukelmann Matar