

PARADIGMAS DE REPRESENTAÇÃO DE MODULAÇÃO DE ALVENARIAS EM FERRAMENTAS BIM

REPRESENTATION PARADIGMS FOR MASONRY MODULATION IN BIM TOOLS

 10.4237/gtp.v4i2.101

Ari MONTEIRO

Mestrando da Escola Politécnica da USP.

| ari.monteiro@poli.usp.br |

| <http://lattes.cnpq.br/0929461339415273> |

Rita Cristina FERREIRA

DWG arquitetura e sistemas s/c, MSC.

| rita@dwg.arq.br |

| <http://lattes.cnpq.br/8816822206656848> |

Eduardo Toledo SANTOS

Prof. Dr. da Escola Politécnica da USP

| eduardo.toledo@poli.usp.br |

| <http://lattes.cnpq.br/8615127367466231> |

RESUMO

As novas ferramentas CAD que suportam o conceito de Building Information Modeling (BIM) baseiam-se na modelagem 3D da geometria do edifício instanciando objetos contidos em famílias de componentes. Alguns destes objetos possuem geometria bastante detalhada, enquanto outros se restringem às suas fronteiras mais externas. No caso do objeto “parede”, esta representação normalmente limita-se às suas faces externas e uma lista de camadas utilizadas para representar sua composição interna (núcleo e revestimentos). Esta representação torna o arquivo leve, favorecendo o desempenho da aplicação. No entanto, para projetos que necessitam de um nível de detalhe superior àquele oferecido por esta solução, como o projeto para produção de vedações verticais em alvenaria (PPVVA), uma representação 3D completa dos elementos da parede é importante. Neste cenário, surge uma questão: como representar os elementos da parede de maneira a atender aos requisitos do PPVVA e, ao mesmo tempo, não degradar o desempenho de manipulação do modelo BIM? Este trabalho apresenta algumas abordagens para a representação da modulação: uma explícita (famílias de objetos) e outras implícitas (array / modelagem generativa). Uma análise comparativa destas abordagens é apresentada, destacando-se as suas vantagens e desvantagens, tais como dificuldade de implementação, facilidade de utilização e impacto no desempenho da aplicação. A viabilidade de cada método foi estudada, restringindo-se ao software Autodesk Revit® Architecture 2009/2010, considerando-se os processamentos para geração de vistas 2D e quantificação de componentes, bem como consumo de memória e processamento ligados a cada abordagem, permitindo a tomada de decisão no nível de implementação.

Palavras-chave: Projeto para produção de alvenaria, parametrização, BIM.

ABSTRACT

New CAD tools which support the BIM (Building Information Modeling) concept are based on 3D-modeling buildings by instancing objects from component families. Some of these objects have detailed geometry while others are restricted to their outer boundaries. In the “wall object” case, this representation usually is limited to its external faces and a list of layers is used to represent its internal composition (core and finishes). This representation makes the file light, helping the application performance. However, in designs that require a higher detail level than this solution supports, like the masonry design for production (MDP), a complete 3D representation of wall components is important. In this context, a question arises: how to represent the wall elements in a way that fulfills the MDP requirements and, at the same time, do not degrade the handling performance of the BIM model? This work presents some approaches to representing masonry modulation: an explicit one (object families) and some others implicit (array / generative modeling). A comparative analysis is presented, highlighting their pros and cons like implementation complexity, ease of use and performance impact on the application. The feasibility of each method was studied on Autodesk Revit® Architecture 2009/2010 considering processing for generating 2D views and quantity take off, as well as memory and CPU consumption for each approach, enabling decision making at the implementation level.

Keywords: Masonry Design for Production, parametrization, BIM.

1. INTRODUÇÃO

Os objetos AEC disponibilizados nas ferramentas BIM possuem níveis de detalhe variados. Alguns destes objetos apresentam geometria bem detalhada, enquanto outros se restringem ao seu volume, suprimindo detalhes da sua composição.

Um exemplo é o objeto *wall* (parede) cuja representação gráfica está limitada às suas dimensões externas. A composição da parede também é representada a partir de uma lista de camadas que definem as características do seu núcleo e de eventuais revestimentos.

Esta representação padrão resulta em arquivos mais leves, que favorecem o desempenho da aplicação. Entretanto, observa-se que, para atender projetos para produção como o PPVVA, é exigido um nível de detalhe superior ao oferecido nas ferramentas BIM disponíveis hoje no mercado.

A exigência de uma representação 3D mais completa destes elementos no PPVVA advém da intensa atividade de compatibilização entre subsistemas presente neste tipo de projeto e da necessidade de projetar-se a composição das fiadas, definindo-se as sequencias de blocos e juntas em cada uma. Infelizmente, incrementar o nível de detalhe dos objetos 3D pode acarretar a diminuição do desempenho da aplicação já que até uma parede de dimensões medianas é composta por centenas de blocos.

Segundo Eastman (2006), a definição de um modelo de edifício no nível de construção, isto é, com um alto nível de detalhe, é um complicado empreendimento que requer a definição e gerenciamento de milhões de objetos.

Este trabalho tem como objetivo responder à seguinte questão: como representar os componentes da parede de maneira a atender aos requisitos do PPVVA e, ao mesmo tempo, não degradar o desempenho de manipulação do modelo BIM?

Para isto foram estudadas as necessidades de representação para os elementos de uma parede no contexto do PPVVA (sessões 2 e 3) e duas alternativas de representação para estes elementos (sessão 4). Na sessão 5 é proposta a utilização de uma gramática para auxiliar a implementação de uma das alternativas de representação.

Foram executados experimentos de aplicação (sessão 6) para as alternativas de

representação explícita propostas usando o Revit® Architecture (versões 2009 e 2010) que é um dos produtos da solução BIM fornecida pela Autodesk Inc. (AUTODESK, 2008a). Após estes experimentos, os resultados foram analisados (sessão 7) com o objetivo de avaliar (sessão 8) qual das alternativas é a mais adequada para o PPVVA.

2. O PROJETO DE ALVENARIA

O PPVVA nasceu no fim da década de 80. Uma das iniciativas foi a partir de um trabalho de pesquisa da EPUSP (Escola Politécnica da Universidade de São Paulo) em convênio com a Encol (empresa construtora) (SILVA, 2003, p. 42-46).

O foco do PPVVA é a racionalização dos processos produtivos e a compatibilização dos subsistemas com os quais a alvenaria possui interface, tais como a estrutura, instalações elétricas e hidráulicas, revestimentos entre outros subsistemas.

A natureza deste projeto conduz o projetista a verificar as interferências entre os diversos sistemas que compõem um edifício, além de demandar a produção de uma precisa documentação para execução.

A alvenaria consiste em dispor pedras, tijolos, blocos, etc., com argamassa ou não (CORONA; LEMOS, 1972, p. 37), incluindo diversos outros elementos, tais como ferro para reforço, graute, vergas, contravergas etc. (CHING, 1999).

Segundo Silva (2003, p. 96), os elementos básicos de uma alvenaria de vedação são as unidades de alvenaria (tijolos ou blocos) e as juntas de argamassa. Neste artigo, identificou-se que estes elementos podem ser agrupados em três categorias específicas:

Elementos básicos: blocos, pedras, tijolos;

Elementos de junção: juntas (que podem ser preenchidas ou não com material aglutinante);

Elementos de estruturação: vergas, contravergas, telas, etc.

Os elementos básicos representam a maior parte da composição da parede, sendo estas partes ligadas entre si por juntas. A função das juntas é conferir estabilidade própria aos elementos básicos, bem como garantir outros fatores de desempenho, tais como isolamento acústico e térmico, impermeabilização, etc. Os elementos de estruturação têm a função de conferir desempenho adequado à

parede nas ligações com outros subsistemas e/ou componentes, tais como portas, janelas, vigas, lajes, pilares, outras paredes, etc.

As juntas de argamassa podem ser classificadas de acordo com sua posição e função na modulação em: a) Juntas horizontais (de assentamento, de fixação e intermediárias) e b) Juntas verticais (secas e preenchidas).

Silva (2003, p. 92), enumera diversas normas técnicas (NBR-5731/82, NBR-5706/97, NBR-10837/89, entre outras) utilizadas como base para o PPVVA, mas também afirma que as práticas vigentes são uma mescla destas normas com a experiência dos projetistas da área.

3. REGRAS PARA MODULAÇÃO DE ALVENARIA

A modulação de alvenaria (Figura 1) é uma atividade complexa que envolve várias regras e diversas variáveis de projeto. O processo de modulação de alvenaria no PPVVA é dividido em três atividades principais. São elas: a) modulação horizontal; b) amarração entre paredes e c) modulação vertical.

A modulação horizontal consiste na distribuição otimizada dos módulos de uma família de blocos ao longo do comprimento da parede. Este processo tem como objetivo gerar as duas primeiras fiadas da modulação de alvenaria.

A amarração entre paredes é uma atividade que define como as paredes se ligam. Em ABCI (1990), podem ser vistos diversos métodos de amarração, como a amarração por tela e por intertravamento. Já a modulação vertical consiste na replicação da modulação horizontal no sentido da altura da parede.

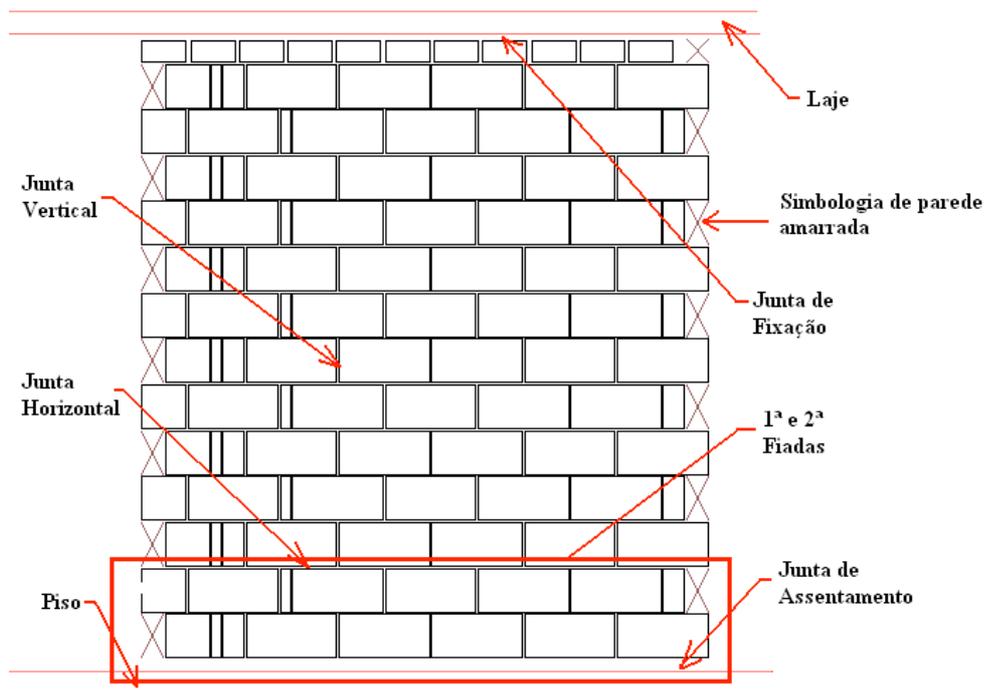


Figura 1 - Elementos básicos de uma modulação de alvenaria.

Em todos estes processos, o projetista deve atentar para a resolução de eventuais interferências entre a alvenaria e outros subsistemas, orientando-se pelas seguintes regras básicas:

Modulação horizontal

Uma fiada pode iniciar com qualquer módulo disponível na família de blocos, entretanto, é mais racional iniciar-se com blocos inteiros;

As juntas verticais podem ser de dois tipos: secas ou preenchidas;

Em uma fiada é possível coexistir juntas verticais secas e preenchidas;

Quando é definida a junta vertical seca, as duas primeiras juntas nas extremidades da fiada devem ser preenchidas sempre;

As juntas secas podem ter espessura variando de 0.3 cm a 0.7 cm;

As juntas preenchidas podem ter espessura variando de 0.8 cm a 1.2 cm;

Utilizar para o cálculo inicial de fiadas a espessura 0.5 cm para juntas secas e 1.0 cm para juntas preenchidas, de forma evitar a utilização de peças de compensação e/ou enchimentos;

Deve-se evitar que as juntas verticais fiquem a prumo, isto é, que as juntas de duas fiadas subsequentes fiquem alinhadas;

Se o cálculo de fiadas gerar juntas a prumo e/ou resíduos, sendo estes menores que o menor módulo disponível na família de blocos, deve-se redistribuir este resíduo na espessura das juntas da fiada;

Se, após a execução da regra acima, não for encontrada uma solução ótima deve-se apelar para o uso de peças compensação e/ou enchimentos;

Uma alternativa à regra anterior é a redefinição da tolerância utilizada para cada tipo de junta vertical e o recálculo da fiada.

Amarração entre paredes

Para a amarração por intertravamento, uma parede entra na outra, alternando blocos nas extremidades das fiadas;

Para a amarração com tela, uma parede é unida a outra parede, ortogonal a ela, com 1.0 cm de junta vertical e, a cada duas fiadas a partir da 2ª fiada, são colocadas telas de ligação, dimensionadas conforme a espessura da parede.

Modulação vertical e ajuste da estrutura e níveis entre pisos

As juntas horizontais podem ter espessura variando de 0.8 a 1.2 cm, mas idealmente é de 1.0cm;

As juntas de assentamento e fixação devem ter espessura variando de 2 a 4 cm;

Utilizar para o cálculo inicial da modulação vertical a espessura de 1.0 cm para juntas horizontais e de 3.0 cm para as juntas de assentamento e fixação, de forma evitar a utilização de peças de compensação e/ou enchimentos;

Deve-se evitar a utilização de peças de compensação e/ou enchimentos na última fiada;

Se o cálculo de modulação vertical gerar resíduos, sendo estes menores que o menor módulo disponível na família de blocos, deve-se redistribuir este resíduo nas juntas da fiada;

Se, após a execução da regra acima, não for encontrada uma solução ótima deve-se utilizar peças de compensação e/ou enchimentos;

Uma alternativa à regra anterior é a redefinição da tolerância utilizada para cada tipo de junta horizontal e o recálculo da fiada.

4. ALTERNATIVAS PARA REPRESENTAÇÃO

Foram propostas duas alternativas para representação dos elementos do PPVVA: uma explícita e outra implícita. Na representação explícita, todos os componentes são modelados utilizando o conceito de família de objetos disponível no Revit®.

Por outro lado, na representação implícita estes mesmos elementos são modelados usando técnicas de modelagem generativa. A modelagem generativa consiste numa técnica de modelagem procedural que utiliza um conjunto de regras para criar modelos 3D. Por meio destas regras é possível definir algoritmos que representam, implicitamente, modelos geométricos.

O Revit® permite, usando os recursos padrão disponíveis, a representação implícita de alguns objetos. Entretanto, estes recursos se mostraram limitados para atender aos requisitos de representação dos elementos básicos do PPVVA.

Nos tópicos seguintes são apresentados os detalhes das duas alternativas de representação propostas neste trabalho.

4.1 Representação explícita

Para a implementação desta representação foi utilizado o conceito de famílias de objetos. Uma família é um grupo de elementos (2D/3D) que possui um conjunto de propriedades comuns (parâmetros) e uma representação gráfica. Os parâmetros de cada elemento de uma família podem assumir diferentes valores. Estas variações dentro da família são chamadas de tipos ou tipos da família (AUTODESK, 2008b; AUTODESK, 2009b).

Para utilizar uma família, inicialmente esta deve ser carregada no projeto. Uma vez carregada, os tipos desta família podem ser instanciados no projeto por meio de um comando específico do Revit®. A partir daí, cada instância pode ter seus parâmetros alterados para atender a necessidades específicas do projeto. Em Autodesk (2009c), estão descritos 3 tipos de famílias:

Famílias de sistema

Definem os elementos básicos da construção: paredes, telhados, tetos/forros, pisos. Elementos que definem configurações do sistema, tais como: níveis, grades, formatos de folha e viewports, também são cobertos por famílias de sistema;

São predefinidas e não é possível alterar suas definições básicas (definição de novos parâmetros). A única customização permitida é a adição de novos tipos numa família existente;

Geralmente não necessitam de outros objetos na tela para instanciar seus tipos.

Famílias carregáveis

São usadas para definir componentes da construção que normalmente são comprados, fabricados ou instalados, ou elementos de anotação.

Estas famílias são criadas em arquivos externos (com extensão .rfa) que devem ser carregados no projeto.

Famílias locais

Definem elementos considerados específicos para um determinado projeto.

A geometria de objetos construídos com famílias locais pode ser associada a outros objetos no projeto (paredes, lajes, telhados, etc.).

Quando os objetos de referência sofrem alterações, estas são propagadas para o objeto da família local.

Famílias locais não podem ser compartilhadas com outros projetos. Elas são sempre criadas no contexto do projeto corrente.

Não é recomendado criar muitas famílias locais em um projeto, pois isto pode degradar o desempenho do Revit.

O tipo de família escolhido para representar os elementos do PPVVA foi a família carregável, pois com ela é possível definir parâmetros customizados e reutilizar estas famílias em vários projetos.

Para atender aos requisitos do PPVVA as seguintes famílias de objetos são necessárias: módulos de blocos, vergas, contravergas, telas e caixilhos. Para simplificar o problema, este artigo foca apenas a representação dos módulos de blocos utilizados para compor a modulação de uma parede. As juntas entre os blocos foram representadas como parâmetros da família, ao invés de elementos 3D.

O primeiro passo no processo de implementação desta alternativa foi a definição da família de blocos que seria modelada no Revit®. Como a idéia era apenas representar qualquer família de blocos, decidiu-se pelos blocos de

concreto de um fabricante tradicional, cujas especificações foram coletadas em seu site (GRESKA, 2008).

Após a definição da família de blocos, passou-se à modelagem desta família no Revit®. Foram consideradas, para este trabalho, apenas as dimensões externas dos módulos de blocos. As furações não foram representadas. Embora o Revit® permita perfeitamente a representação completa dos blocos, foram adotadas estas simplificações que não impactaram muito nos resultados encontrados.

Para criar uma família, o software dispõe de diversos *templates* (arquivos modelo). A escolha do arquivo modelo depende de como os tipos da família deverão interagir com os outros elementos do projeto (AUTODESK, 2008c; AUTODESK, 2009c).

No caso específico dos blocos de alvenaria, a ideia é que eles fiquem hospedados em objetos parede. Entre os arquivos modelos (*templates*) disponíveis, o que atendeu a estas especificações foi o *Metric Generic Model wall based.rft*.

Este tipo de modelo é utilizado para criar famílias de objetos cujos tipos podem ser instanciados apenas dentro de paredes. Ao iniciar um novo projeto de família usando este *template* uma parede exemplo é carregada. Esta parede serve como referência para modelagem da nova família.

Outra característica interessante da utilização deste *template* é que algumas dimensões do objeto definido na família podem se ajustar em função das dimensões da parede. Para que isto ocorra, durante o processo de modelagem da família o usuário deve restringir a geometria do novo objeto às faces da parede de referência. Neste processo, as partes da geometria restringidas à parede de referência não devem receber cotas, pois do contrário, estas partes não acompanharão a atualização das dimensões da parede no projeto.

Considerando esta característica, o parâmetro da espessura do bloco foi vinculado à espessura do núcleo da parede, ficando os parâmetros comprimento e altura do bloco definidos pelos tipos da família.

O problema visualizado neste ponto foi que, ao utilizar esta família de blocos em um projeto que possua paredes com diversas espessuras, o comando para gerar quantitativos não faria distinção entre blocos do mesmo tipo, mas com espessuras diferentes. Por exemplo, blocos inteiros em paredes de 14 cm e 9 cm,

seriam contabilizados juntos, quando na verdade não deveriam.

A solução adotada neste trabalho foi a duplicação da família de blocos, considerando as seguintes diferenças entre elas:

O nome das famílias contém a espessura da parede para qual elas são aplicadas;

Os tipos serão nomeados com o modelo do bloco e todas as dimensões do bloco (comprimento x altura x espessura).

A Figura 2 apresenta os parâmetros de controle definidos para uma família de blocos específica para paredes de 14 cm de espessura.

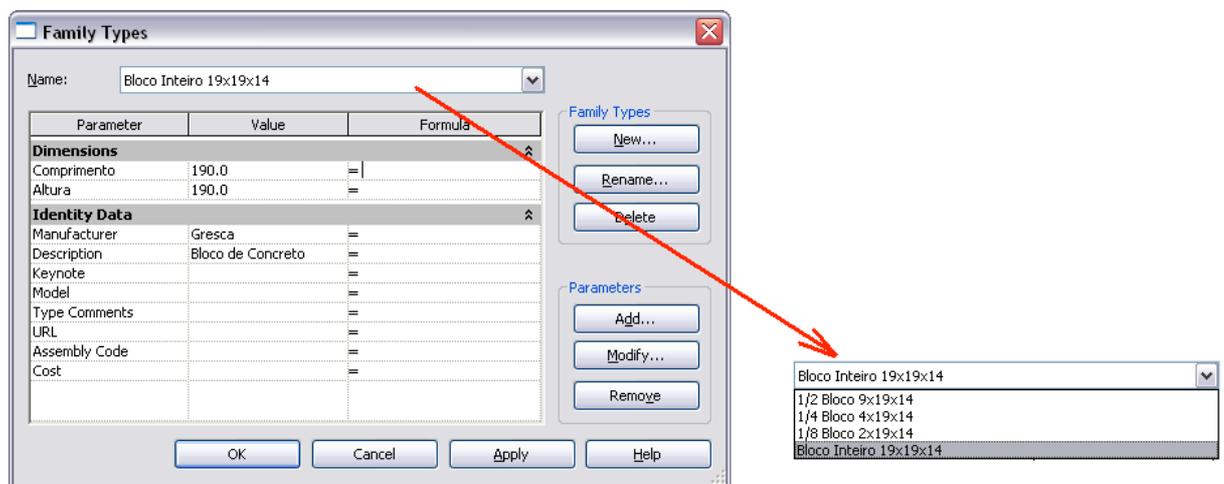


Figura 2 - Tipos da família de blocos Gresca para paredes de 14 cm.

O Revit® possui o conceito de *nested families* (famílias aninhadas). Este conceito permite a criação de famílias que incluem outras famílias. Utilizando este conceito, foi criada uma família chamada *alvenaria*, onde tipos da família de blocos foram instanciados na forma de um *array* parametrizado.

Arrays são conjuntos de elementos que se repetem. Os *arrays* lineares são distribuídos sobre uma linha reta e os radiais/polares sobre um arco. No comando de criação de *arrays*, o número de elementos é sempre um dos parâmetros. O outro parâmetro é o espaçamento entre os elementos ou o comprimento/ângulo total. No caso dos *arrays* paramétricos, os elementos são sempre associados e os parâmetros podem ser editados, de forma que a movimentação de um elemento automaticamente altera o espaçamento entre os demais, mantendo-o uniforme. A alteração no número de elementos provoca sua redistribuição no comprimento total, se este for um dos parâmetros.

Na família *alvenaria* foram definidos parâmetros para controlar o número de

itens do *array* paramétrico em função do comprimento e da altura da parede. Segundo Ferreira (2007, p. 50), rotinas AutoLISP™ podem ser utilizadas para automatizar a atividade de modulação no AutoCAD®. O acesso a estas rotinas viabilizou o uso de alguns dos parâmetros necessários na concepção da família *alvenaria* (Figura 3).

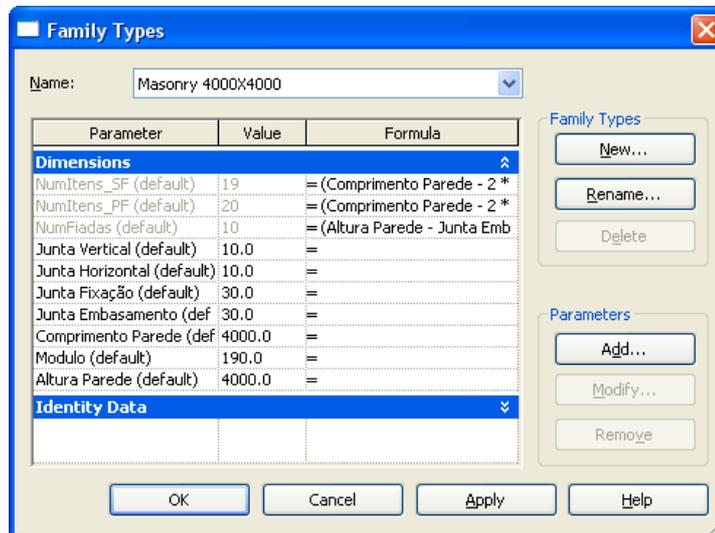


Figura 3 - Parâmetros da família alvenaria.

Os parâmetros definidos para controlar o número de elementos do *array* paramétrico têm seus valores controlados por fórmulas que, por sua vez, utilizam os demais parâmetros da família *alvenaria*.

Os parâmetros Comprimento Parede e Altura Parede foram criados porque não foi encontrada na documentação disponível do Revit® 2009, uma forma de referenciar estes parâmetros nas fórmulas (AUTODESK, 2008b; AUTODESK, 2008c; AUTODESK, 2009c). Também foi pesquisada esta funcionalidade na documentação do Revit® 2010 e verificou-se que este problema continua existindo.

O ideal seria que, ao instanciar um tipo da família alvenaria sobre uma parede do projeto, fosse possível captar automaticamente desta os valores correspondentes ao seu comprimento e altura. Sendo assim, para utilizar esta família, o projetista deverá atribuir valores a estes (comprimento e altura) e aos demais parâmetros após a instanciação do tipo.

Quando tipos de uma família aninhada são instanciados, seus elementos internos não podem ser acessados. Neste caso os blocos não podem ser contabilizados pelo comando de geração de quantitativos e os tipos da família

não podem ser instanciados (AUTODESK, 2009c).

Entretanto, o Revit® possui o conceito de *shared families* (famílias compartilhadas). Este conceito permite habilitar o acesso aos tipos das famílias utilizadas para compor uma família aninhada. A família de blocos foi adaptada para incorporar o conceito de famílias compartilhadas, utilizando o comando *Settings > Family Category and Parameters*.

4.2 Representação Implícita

A representação implícita propõe que não sejam modelados os blocos, as juntas e demais elementos da parede. Ao invés de modelar estes elementos de forma independente, a idéia desta proposta é que uma família de parede especial seja criada.

Esta nova família deverá incorporar a representação dos blocos e juntas por meio de um *array* parametrizado. Já a representação dos elementos de estruturação ficaria a cargo de famílias independentes que teriam seus tipos instanciados sobre paredes desta família especial. A família de parede especial deverá também regenerar automaticamente o *array* paramétrico após a instanciação de um tipo pertencente a uma família de interface (vergas, contravergas, caixilhos).

No Revit®, os objetos parede são definidos a partir de famílias especiais chamadas famílias de sistema. Famílias deste tipo não podem ser criadas e sua definição é uma “caixa-preta”, isto é, não há como acessar sua implementação (AUTODESK, 2008c). O máximo de customização permitida para famílias de sistema é a criação de novos tipos, que sempre respeitarão as definições (parâmetros) originais da família (AUTODESK, 2009c).

Consultando a documentação disponível para gerenciamento de famílias do Revit® 2010, verificou-se que mesmo na nova versão não é possível a criação de novas famílias de sistema. Também foi consultada a documentação para desenvolvimento de customizações (API) e nela consta que não há maneiras de criar novas famílias de sistema usando os recursos de programação.

A criação de novos tipos dentro de uma família de parede permite a criação de novas camadas (Figura 4) com suas características, mas não permite a criação de novos parâmetros, que seriam úteis para a implementação de um *array* paramétrico.

Na definição das camadas de uma parede é possível representar blocos usando uma hachura. Inclusive já existem famílias com estas características (AUTODESK, 2008b), mas não foi possível representar com hachura os vários módulos de uma família de blocos. Além disso, uma modulação pode assumir diversas configurações diferentes em função dos elementos de interface e elementos pertencentes a outros subsistemas, como os de instalações, por exemplo.



Figura 4 - Definição de camadas em um tipo da família Basic Wall.

Considerando as limitações apresentadas para a concepção da família de parede especial, partiu-se para a avaliação da utilização de técnicas de modelagem generativa como abordagem de solução para este problema.

A utilização de modelagem generativa pressupõe a definição de vocabulários de formas e de regras para representação de objetos. O conjunto definido pelas regras de representação e pelo vocabulário de formas é chamado de gramática de forma.

Utilizando famílias carregáveis, os recursos disponíveis na API do Revit® e o conceito de gramáticas, pretende-se implementar uma família especial que terá como objeto hóspede paredes convencionais.

A idéia é que esta família represente a modulação de alvenaria e armazene as variáveis utilizadas para gerar esta modulação. Uma expressão compacta armazenada numa propriedade desta família representaria a modulação projetada. Regras de modulação traduziriam esta expressão numa representação gráfica 2D (textura/símbolo), de acordo com a necessidade de visualização..

Os resultados do processamento desta expressão poderiam ser: a) uma textura

aplicada nas faces do objeto parede; b) representação simbólica das fiadas em planta e elevação; c) a quantidade de módulos de bloco e o volume de argamassa consumido pelas juntas, ambos os valores armazenados dentro de propriedades da parede customizada.

Adicionalmente, deve ser possível atualizar esta textura e quantitativos quando as dimensões do objeto parede hóspede forem alteradas pelo usuário. Acredita-se que este comportamento possa ser implementado relacionando os identificadores da parede hóspede e da parede customizada, por meio dos recursos da API do Revit®.

5. GRAMÁTICA PARA REPRESENTAR MODULAÇÕES

A idéia de utilizar uma expressão compacta para representar a modulação de uma parede remete à necessidade de definir formalmente quais elementos (símbolos (formas) e palavras reservadas) deverão compor esta expressão ou, em outras palavras, qual deve ser a sintaxe desta expressão. Para definir esta sintaxe, uma gramática pode ser usada. As gramáticas são formalismos utilizados para definir linguagens simbólicas ou visuais.

Estudos iniciais sobre a especificação de gramáticas de forma permitiram esboçar uma gramática especializada para representação de modulações de alvenaria.

Segundo Celani et al. (2008), os elementos essenciais de uma gramática de forma, os quais devem ser definidos nesta ordem, são:

Vocabulário de formas – conjunto finito de formas primitivas que podem ser bidimensionais ou tridimensionais;

Relações espaciais – conjunto de combinações espaciais desejadas entre as formas primitivas do vocabulário;

Regras – a partir das relações espaciais são definidas regras de transformação do tipo $A \rightarrow B$ (ao encontrar A, substitua por B). Estas regras podem ser classificadas em três grupos: adição, substituição e subtração;

Forma inicial – para dar início a aplicação das regras, deve-se selecionar uma forma inicial, pertencente ao vocabulário de formas.

Seguindo estas diretrizes, somadas àquelas apresentadas na sessão 3 deste trabalho, foram definidos os elementos essenciais de uma nova linguagem

visual, que denominamos MML (*Masonry Modulation Language*). A Tabela 1 descreve os elementos da gramática que gera a MML.

O escopo da MML é a descrição da modulação de alvenarias utilizando para tal uma gramática de forma que contém as regras e as formas básicas de uma modulação (blocos e juntas). Estão fora do escopo da MML a representação de elementos de interface com a alvenaria, tais como vergas, contra-vergas, telas e caixilhos.

Elemento	Descrição
Vocabulário de formas	Módulos de blocos (bloco inteiro, 1/2 bloco, 1/4 bloco, 1/8 bloco); Juntas Verticais (secas e preenchidas); Juntas Horizontais (assentamento, fixação e intermediárias).
Relações espaciais	Posicionamento horizontal: módulo de bloco + junta vertical; Posicionamento vertical: módulo de bloco + junta horizontal; Posicionamento horizontal com rotação do módulo a 90 graus: módulo de bloco rotacionado + junta vertical.
Regras	As regras de transformação utilizadas na MML foram agrupadas em duas categorias: Aditivas – encarregadas pela justaposição de módulos de bloco com juntas verticais (modulação horizontal) e de fiadas com juntas horizontais (modulação vertical); Substitutivas – encarregadas pela substituição de módulos não rotacionados por uma versão rotacionada. Situação típica de finalização da modulação vertical. Também estão incluídas nesta categoria as regras para amarração de paredes, pois estas envolvem a troca de módulos de bloco nas extremidades das fiadas.
Forma inicial	Qualquer módulo de bloco poderá ser utilizado como forma inicial.

Tabela 1 - Elementos da gramática MML.

6. EXPERIMENTOS DE APLICAÇÃO

Os resultados apresentados nesta sessão restringem-se apenas a solução para representação explícita, baseada na utilização de famílias de objetos. A solução para representação implícita, ainda não foi implementada, pois está em fase de especificação.

Utilizando as famílias de objetos descritas na sessão anterior, foram executados experimentos com o objetivo de avaliar qual das famílias propostas é a mais adequada para o PPVVA. O Quadro 1 descreve as configurações do computador utilizado para realizar os experimentos.

Modelo	NOTEBOOK ACER ASPIRE 4520
Processador	AMD ATHLON X2 64 BITS
Memória RAM	2 GB
Memória Virtual	2 GB
Disco Rígido	TOSHIBA MK 1646GSX 150 GB
Adaptador de Vídeo	NVIDIA GEFORCE 7000M/NFORCE 610M 512 MB
Sistema Operacional	WINDOWS XP PROFESSIONAL SP3 32 BITS

Quadro 1 - Configuração do computador utilizado

Nestes experimentos, foi considerada a modulação de uma parede “cega”, isto é, sem aberturas e elementos estruturais. Embora este tipo de parede seja uma exceção no PPVVA, ela se mostrou eficiente para chegar a algumas conclusões acerca do problema abordado neste artigo.

6.1 Experimento 1 – Família de Blocos

Neste experimento foi utilizada a família de blocos criada para executar a modulação de uma parede. Abaixo, estão descritos os passos deste experimento:

Arquivo de parede:

Criação de um novo arquivo de projeto contendo uma parede da família Basic Wall: Interior Blockwork 140, com 4 metros de comprimento e 4 metros de altura;

Carregamento da família de blocos, criada previamente, no projeto;

Instanciação de dois blocos inteiros e um meio bloco para compor os elementos iniciais da primeira e segunda fiadas;

Posicionamento destes blocos utilizando o comando *Dimension*;

Replicação dos blocos da primeira fiada usando o comando *Array* com opção *Group and Associate* marcada. Esta opção permite associar os elementos do *array* e agrupá-los, tornando o *array* paramétrico;

Replicação dos blocos da segunda fiada usando o comando *Array* com opção *Group and Associate* marcada;

Instanciação de um meio bloco para finalizar a segunda fiada;

Replicação da primeira e segunda fiadas, no sentido da altura da parede, usando o comando *Array* com opção *Group and Associate* marcada.

Cópia do arquivo de parede:

Criação de três cópias do arquivo de parede usando o comando *File > Save As*.

Arquivo mestre:

Criação de um novo projeto para agrupar as cópias das paredes;

Inserção de referências das paredes usando o comando *File > Import/Link > Revit*;

Cópias das referências dentro do projeto usando os comandos *Copy* e *Mirror*.

Quantificação de blocos:

Extração de quantitativo de blocos usando o comando *Schedule/Quantities* no painel *View*.

Plantas de primeira e segunda fiadas:

Extração de plantas de primeira e segunda fiadas usando os comandos de geração de vistas do Revit (Figura 5).

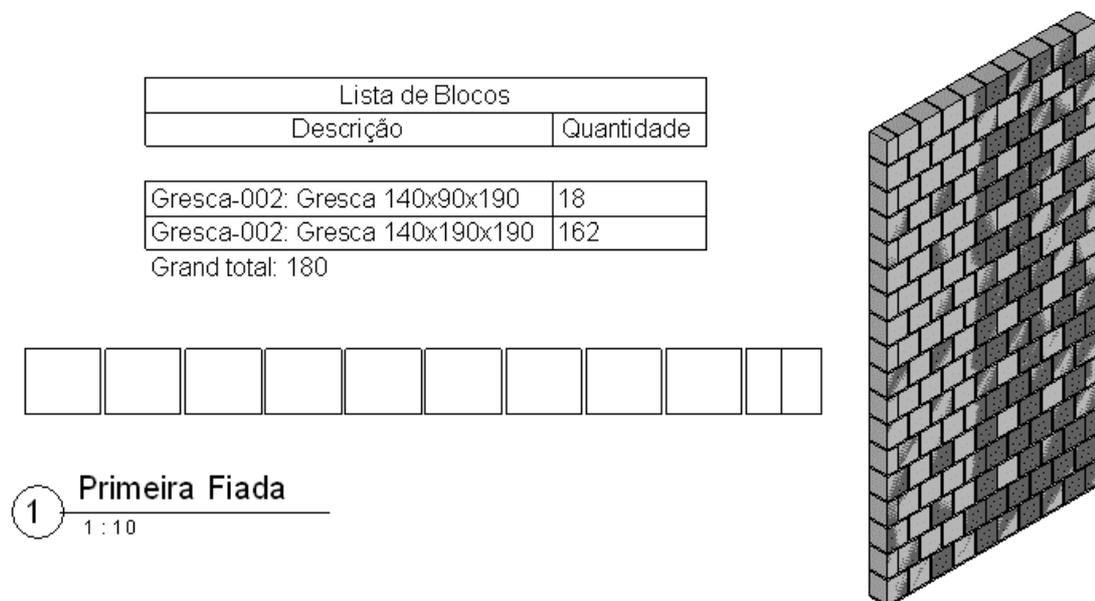


Figura 5 - Planta de primeira fiada e quantitativo de blocos extraídos do modelo 3D de uma parede construída com a família de blocos desenvolvida.

6.2 Experimento 2 – Família Alvenaria

Neste experimento foi utilizada a família *Alvenaria* (compartilhada e aninhada)

para executar a modulação de uma parede. Esta família contém instâncias da família de blocos dispostos na forma de um *array* paramétrico. Abaixo, estão descritos os passos deste experimento:

Arquivo de parede:

Criação de um novo arquivo de projeto contendo uma parede da família *Basic Wall: Interior Blockwork 140*, com 4 m de comprimento e 4 metros de altura;

Carregamento da família Alvenaria no projeto;

Instanciação do tipo padrão da família Alvenaria;

Edição dos parâmetros do tipo instanciado (juntas, comprimento e altura da parede) para regeneração do *array* paramétrico.

Cópia do arquivo de parede:

Idem experimento 1.

Arquivo mestre:

Idem experimento 1.

Quantificação de blocos:

Idem experimento 1.

Plantas de primeira e segunda fiadas:

Idem experimento 1.

Também foi estudada a alternativa de criar uma família para representar fiadas, ao invés de uma modulação completa da parede. O objetivo foi verificar se a utilização desta família deixaria a manutenção das fiadas mais prática. As rotinas AutoLISP™ utilizadas por Ferreira (2007) utilizam regras para distribuição de blocos numa fiada, mas não foi possível incorporar estas regras utilizando fórmulas na definição da família proposta pois o recurso de fórmulas é muito simples se comparado aos recursos de programação disponíveis na linguagem AutoLISP™. O Revit não permite programar novos comandos usando fórmulas em famílias. As regras contidas nas rotinas usadas por Ferreira (2007) não só calculavam as juntas dos blocos, mas também selecionavam os módulos de bloco mais adequados em função do comprimento da parede. Nestas rotinas AutoLISP™, o *array* paramétrico era montado em tempo de execução para se adaptar aos diversos comprimentos de parede. No

caso do Revit[®], o mesmo *array* está predefinido na família alvenaria.

7. ANÁLISE

7.1 Experimento 1 – Família de Blocos

Os arquivos gerados neste experimento utilizaram os comandos padrão do Revit[®] (*Array* e *Dimension*) para distribuição dos blocos. O posicionamento e edição de cada bloco mostraram-se bastante flexíveis. Em cada fiada foi possível modificar o número de itens do array e trocar os módulos dos blocos quando necessário.

Quando o comando *Array* é utilizado, o Revit[®] cria um grupo de objetos se a opção *Group and Associate* for habilitada (AUTODESK, 2008b). Após o comando é possível alterar o número de elementos sem desagrupar o array, mas se for necessário trocar algum bloco dentro do *array*, o desagrupamento precisa ser feito, perdendo-se a parametrização do *array*.

Como relação ao tempo de resposta nos comandos de visualização (*Rotate*, *Zoom*, *Pan*), verificou-se que no arquivo de parede o desempenho do Revit[®] foi bom. Mas, na visualização do arquivo mestre contendo 16 paredes cegas, sendo 4 delas referências e demais cópias destas referências, o desempenho de regeneração caiu um pouco com relação ao caso anterior. Também foram contabilizados os tamanhos dos arquivos gerados neste experimento (Tabela 2).

Descrição	Espaço Disco	em	Espaço em Memória RAM (aprox.)
Família de blocos Gresca	196 KB		47 KB
Arquivo de Parede (140x4000x4000 mm)	676 KB		75 KB
Arquivo mestre com 16 paredes	1988 KB		117 KB

Tabela 2 - Tamanho dos arquivos no experimento 1.

7.2 Experimento 2 – Família Alvenaria

Neste experimento, o posicionamento dos blocos foi automatizado pelo *array* parametrizado incorporado no tipo padrão da família alvenaria. O posicionamento da modulação predefinida na família alvenaria foi executado com o comando *Dimensions* do Revit[®].

Depois de instanciar o tipo padrão da família alvenaria, o projetista de vedações precisa atualizar os parâmetros da instância para regenerar o array paramétrico. Verificou-se que o tempo de regeneração do array foi muito alto, chegando a demorar, em alguns casos, vários minutos, dependendo das dimensões da parede (comprimento e altura).

Nos comandos de visualização (*Rotate, Zoom, Pan*) e tempo de resposta foi mais rápido na manipulação do arquivo mestre, comparado com o arquivo mestre gerado no experimento 1. O Revit® otimizou a exibição das paredes compostas pela família alvenaria. O tamanho dos arquivos gerados no experimento 2 estão relacionados na Tabela 3.

Descrição	Espaço em Disco	em Espaço em Memória RAM (aprox.)
Família de blocos Gresca	196 KB	47 KB
Família alvenaria	504 KB	67 KB
Arquivo de Parede (140x4000x4000 mm)	1952 KB	75 KB
Arquivo mestre com 16 paredes	684 KB	103 KB

Tabela 3 - Tamanho dos arquivos no experimento 2.

8. CONCLUSÕES

No PPVVA, uma alvenaria pode assumir diversas configurações em função das necessidades específicas de compatibilização com os subsistemas e ou elementos (verga, contraverga etc.) com os quais faz interface. Para viabilizar a compatibilização entre diversos subsistemas, a representação dos elementos de uma parede torna-se crucial.

Para atender a este requisito do PPVVA, o caminho foi adotar uma representação explícita dos elementos da parede utilizando famílias para representar cada uma dos elementos da modulação: blocos de concreto, vergas, contravergas, telas e caixilhos. Embora esta abordagem naturalmente degrade o desempenho da aplicação, ela foi a que ofereceu mais flexibilidade para o projetista de vedações.

A utilização de um *array* paramétrico para representar uma modulação mostrou-se ser uma solução interessante para automatizar o processo de distribuição dos blocos utilizando os recursos padrão do Revit®. Entretanto, esta

solução não ofereceu flexibilidade na edição das fiadas e na resolução automática das diversas possibilidades de modulação disponíveis para uma determinada família de blocos.

Durante o processo de compatibilização, o projetista de vedações precisa com frequência alterar a distribuição dos blocos na parede. Desta forma, mesmo sendo prático, o *array* paramétrico acabará sempre sendo desfeito neste processo. Esta característica do PPVVA acaba tornando a modulação um componente dinâmico e de difícil implementação usando um *array* paramétrico.

Ainda sobre a utilização do conceito de *array* paramétrico, esbarrou-se na dificuldade técnica de incorporar as regras de distribuição de blocos utilizadas por Ferreira (2007) dentro de suas rotinas AutoLISP™. As regras contidas nestas rotinas AutoLISP™ selecionam os módulos de blocos e calculavam as juntas verticais em função do comprimento da parede, de forma otimizada. O recurso de fórmulas utilizado dentro da definição de parâmetros nas famílias não ofereceu suporte para este tipo de customização (AUTODESK, 2008b).

Com relação à implementação da representação implícita dos elementos de uma parede, foi estudado o recurso *custom patterns* (hachuras personalizadas). Verificou-se que é possível criar novos padrões de hachura e aplicá-los nas faces externas das paredes, de forma a simular um revestimento. No entanto, os controles para parametrização de hachuras são muito simples, e estão limitados ao controle do espaçamento e inclinação dos elementos. Também é possível utilizar uma linguagem de descrição para definir novos padrões de hachura, mas nessa linguagem não foi possível descrever toda complexidade de uma modulação.

As limitações encontradas para implementar a representação implícita por meio dos recursos de usuário padrão do Revit® levaram a avaliação do uso de técnicas de modelagem generativa como abordagem de solução para este problema.

As conclusões deste trabalho estão limitadas ao escopo dos recursos disponibilizados no Revit® Architecture 2010, o que não descarta um trabalho de pesquisa similar em outras ferramentas BIM, com objetivo de verificar qual das ferramentas disponíveis no mercado é mais adequada para o PPVVA.

Como propostas de trabalhos futuros ficam o detalhamento da especificação da MML, de forma incluir o tratamento das demais regras de modulação, e a posterior implementação desta linguagem visual numa ferramenta BIM.

REFERÊNCIAS

- ABCI. Associação Brasileira da Construção Industrializada. **Manual técnico de alvenaria**. São Paulo: Projeto/PW editores, 1990.
- AUTODESK. **Building Information Modeling: The Power of BIM**. 2008a. Disponível em: <<http://www.autodesk.com/bim>>. Acesso em: 25 mar. 2008.
- AUTODESK. **Revit Architecture 2009: User's Guide**. 2008b.
- AUTODESK. **Revit Architecture 2010: User's Guide**. 2009a.
- AUTODESK. **Melhores práticas para a criação de componentes paramétricos (famílias) com o Autodesk Revit**. 2008c. 15p. Disponível em: <<http://www.autodesk.com/bim>>. Acesso em: 15 out. 2008.
- AUTODESK. **Revit Architecture 2009: Families Guide – Metric Tutorials**. 2009b. 812p. Disponível em: <<http://www.autodesk.com/revitarchitecture-documentation>>. Acesso em: 05 jan. 2009.
- AUTODESK. **Revit Architecture 2010: Families Guide – Metric Tutorials**. 2009c. 812p. Disponível em: <<http://www.autodesk.com/revitarchitecture-documentation>>. Acesso em: 15 set. 2009.
- CELANI, G.; CYPRIANO D.; GODOY G.; VAZ C.E. **A gramática da forma como metodologia de análise e síntese em arquitetura**, 19p, 2008. Disponível em: <http://www.ufpel.edu.br/faurb/prograu/documentos/artigo2-sustentabilidade.pdf>. Acessado em: 25/09/09.
- CHING, F. D. K. **Dicionário Visual de Arquitetura**. São Paulo: Ed Martins Fontes, 1999.
- CORONA, E.; LEMOS, C. **Dicionário da Arquitetura Brasileira**. São Paulo: Edart, 1972.
- EASTMAN, C. **New Opportunities for IT Research in Construction**. In: SMITH, I.F.C. EG-ICE 2006/ LNAI 4200. Berlin:Springer, 2006. p. 163-174.
- FERREIRA, R. C.. **Uso do CAD 3D na compatibilização espacial em projetos de produção de vedações verticais em edificações**. 2007. 160 p. Dissertação (Mestrado). Escola Politécnica da Universidade de São Paulo, São Paulo, 2007.
- GRESKA. **Vedação 39 cm**. 2008. Disponível em: <<http://www.ceramicagresca.com.br>>. Acesso em: 20 set. 2008.
- SCHEER, S.; AYRES FILHO, C.; AZUMA, F.; BEBER, M. **CAD-BIM requirements for masonry design process of concrete blocks**. In: CIB W78 INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY IN CONSTRUCTION, 25., 2008, Santiago. **Proceedings...** Santiago:Universidad de Talca/Stanford University, 2008, p. 40-47.
- SILVA, M.M.A. **Diretrizes para o projeto de alvenarias de vedação**. 2003, 167p. Dissertação (Mestrado). Escola Politécnica da Universidade de São Paulo, 2003.