Building Information Modeling for Masonry
Phase II Project

Project 3, Masonry Wall Definition Project
Pankow Foundation Grant RGA #02-15

Report 3

Draft Specification for Software Functionality
for BIM-M Modeling Applications

delivered to

Charles Pankow Foundation
in support of the
Building Information Modeling for Masonry Initiative (BIM-M)

Georgia Institute of Technology
School of Architecture
Digital Building Laboratory

T. Russell Gentry, PI
Charles Eastman, co-PI
Andres Cavieres, Graduate Research Assistant

Jamie Davis, Ryan-Biggs | Clark Davis
Project Manager

David Biggs
BIM-M Coordinator

31 August 2015

Contents

EXECUTIVE SUMMARY

The report addresses technical recommendations for the implementation of Building Information Modeling tools for masonry walls. These recommendations are based on the research developed in the Digital Building Laboratory at Georgia Tech during the first semester of 2015. Much of the requirements for the masonry wall project are taken from the BIM-M Benchmark project, which involved discussions with experts from the architecture practice, construction and software industries to understand the information requirements of different stakeholders during conventional design and construction workflows. This study led to the identification of problems and limitations in current BIM systems regarding the representation of masonry walls, in two main aspects.

The first aspect is the lack of a schema to address the description of masonry specific entities, in particular of features and parts that define the shape, structure and the various performance requirements associated to masonry wall assemblies. This schema is necessary to support the development of a wide range of masonry specific applications not currently available to the industry. The second problematic aspect is the lack of proper geometric representation to support the design process of masonry assemblies. More specifically, current BIM systems do not provide the appropriate support to manage the level of geometric information needed at different design stages by different design stakeholders. This aspect includes the lack of geometric operators to create and modify geometric entities according to masonry specific semantics. From this last observation is clear that both aspects are directly related.

To address these issues the research focused first on the theoretical development of a conceptual model or schema to support the representation of masonry walls. The proposed schema described here is based on the concept of masonry regions, as an abstract representation of masonry wall features that are relevant for different stakeholders. The research then proposes a functional classification of different region types as foundation upon which masonry specific applications can be elaborated in the future. These include but are not limited to parametric modeling and rule-checking applications for early design validation, structural and energy analysis, cost estimation and construction planning. It will also provide the basis for the definition of model views necessary for particular data queries and exchanges between design, engineering and construction applications. The document is organized into the following sections: 1) Background, 2) Theoretical Approach, 3) Recommendations and 4) Summary.

We anticipate that this document will be reviewed by our software stakeholders from the BIM community (Autodesk, Bentley, SketchUp, Tekla and Vectorworks) as well as by members of our masonry software community (CAD BLOX, Tradesmen's). We expect the community to comment on the approach provided here, and on the generic and IFC schema described herein which will continue to develop in Phase III of the project. We are continuing to work on an XML version of this schema, which we will provide as an update to this document at the end of August, 2015.

Though we welcome the document review by our BIM-M stakeholder community, we do not expect them to find this document accessible, as it provides a theoretical and data-focused view on the modeling of masonry walls – tailored to the BIM software community that will ultimately implement the BIM-M vision.

1. BACKGROUND

In January 2013, the Digital Building Laboratory (DBL) at the Georgia Institute of Technology completed a road-map to bring Building Information Modeling (BIM) to the North-American masonry industry (Gentry 2013). This overall project involves industry trade associations and stakeholders from throughout the masonry industry, including BIM software vendors and other domain experts from the AEC industry. The road map outlines three phases of research and development. The Development Phase (Phase II) focuses on further elucidating the workflows and software requirements for BIM for masonry (BIM-M), and the completion of three seminal projects that will underpin the BIM-M software specification. These projects include the Masonry Unit Model Definition project (MUMD), the BIM-M Benchmark and the Masonry Wall Definition project (MWD).

The MUMD project focuses on the development and prototyping of a standardized database schema for masonry units and components. The goal is to capture all of the geometric and non- geometric information needed during the development of masonry buildings, from selection, specification and procurement of masonry units, down to the phases of construction planning and execution, operations and maintenance. The intent is that the proposed data model will act as a basis for a series of BIM-integrated services, including digital product catalogs, specialized e-commerce services, cost-estimating among others (Witthuhn et al. 2014; Sharif et al. 2015).

The BIM-M Benchmark project seeks to document how BIM-based processes can improve efficiency and competitiveness of all stakeholders involved in masonry buildings: from architects, to engineers, manufacturing companies, contractors and masons. For that purpose it is necessary to gain a deep understanding on the most common workflows and best practices developed by leading teams in the masonry industry. This understanding covers the entire lifecycle of masonry buildings, from material discovery and selection during early design exploration, structural and environmental analysis, cost estimation, detailing and specification, procurement and delivery, construction planning and coordination, erection and maintenance. The general goal of this project is to formulate specific recommendations on how BIM applications can enhance masonry workflows, bringing value for the industry as a whole (Florez et al. 2014; Gentry et al. 2014). At a more concrete level this means the identification of information requirements for key design activities that provide the basis for the specification of masonry-specific BIM applications (Lee et al. 2015).

The third project, the Masonry Wall Definition project (MWD) addressed in this report focuses on the specification of a process-driven representation for masonry wall assemblies. Such a representation needs to be flexible enough to accommodate the evolution of information requirements from different stakeholders, as the state of the project progresses from early design stages down to construction and operation. As mentioned previously, the information associated to masonry units and other masonry components is being addressed by the MUMD project, while the specification of stakeholder's views along with their domain- specific information requirements has been addressed by the BIM-M Benchmark project. Thus the current MWD capitalizes from these previous projects, in an attempt to provide a comprehensive set of guidelines and specifications for the future implementation of BIM applications for masonry.

2. THEORETICAL APPROACH

This section addresses the theoretical aspects behind the proposed conceptual schema and functionality for BIMM masonry wall applications.

2.1 Top-Down versus Bottom-Up Modeling Approach

The top down modeling approach is the method by which most architects and BIM modelers use to create a building model during design. As the model is created, large sections of walls are defined as blank planes and follow-on activities include the definition of various features and parts, including the creation of openings, the addition of movement joints, structural reinforcement, etc. This top down approach is quite different than the way in which a masonry building is constructed, where walls are made from individual masonry units which are cut, combined with accessories, and then mortared into the wall. Therefore, there exists a theoretical disconnect between the way in which the masonry wall is modeled in BIM applications, and the way in which the wall is physically constructed. For this reason there is a need for a modeling schema that supports both the abstract notion of top-down modeling, as well as the notion of masonry walls as assemblies of smaller modular components. While the geometry of these smaller components may not necessarily be represented in an explicit way, the model still should be able to provide the necessary levels of information to facilitate the realization of the design intent. In practical terms this means that a masonry BIM model should be queryable in relation to different aspects of design and construction.

2.2 Masonry Model Queries

The functionality of masonry-specific BIM applications should be assessed in terms of providing effective support for a range of masonry-related queries and transactions. These may span from simple extraction of geometric properties down to information pertaining to more complex relationships that may exist at the geometric, spatial, temporal or functional level. In the most basic scenario, the information needed is already present in the model, normally in the form of a property value that has been explicitly asserted by the designer, and therefore can be directly extracted from model objects.

In more complex scenarios however, the answer to the query needs to be computed by deriving new information from pre-existent data from different sources and possibly with different formats. It is in these situations where a well formulated data model can become useful, by providing support for the automation of various design tasks. For example, rule-checking procedures can be developed to parse entire models and verify if the current state of the design is compliant with certain requirements, such as building codes, construction standards or best practice guidelines (Eastman et al. 2009). Such techniques have been used to compute whether egress paths in buildings meet fire-code requirements (Lee 2011) or safety measures have been specified for construction activities (Zhang et al. 2013). This type of verification relies in the implementation of querying routines, where the answer for questions of the type "pass" or "fail" is usually implicit in the model and has to be computed on demand.

In the case of masonry, the derivation of new information consists not only in the creation of additional levels of geometric detail, **but more importantly, in the formation of new levels of abstraction.** For example, to evaluate the performance of a masonry component from a given point of view it is not enough to have detailed information about its geometric and material properties. Its performance regarding any engineering perspective can only be evaluated in the context of the larger

system, and under specific scenarios of use. This is because many relationships that are relevant at the system-level are often more abstract than the ones defined at the geometric level, and therefore can only be described with a different set of representations (e.g. time-dependent aspects of the system). For that purpose there must be an underlying conceptual model where more abstract aspects relationships of masonry walls are described in a formal, machine-readable format.

2.3 Masonry Regions

One of the main challenges regarding the implementation of BIM tools for masonry is precisely the abstract nature of many relationships that need to be represented in a masonry assembly model. One example that is relevant in the domain of masonry construction is cost estimation, given certain design parameters of a masonry wall. A simple approach would be to calculate the number masonry courses and the number of masonry units per course. For standard running bond with no reinforcement, the productivity would be assessed by the number of bricks or CMU blocks that masons can typically lay per hour.

However, this approach does not account for situations in which boundary conditions or geometric features of the design deviate from conventional configurations. Additionally, assessments on mason's productivity in relation to design alternatives are rarely formulated in a stand-alone basis. Generally, they are developed in the context of a larger decision-making activity involving some form of trade-off analysis with other kinds of requirements. In this particular case, the construction team may want to know alternative sequences of erection of a masonry wall, in order to assess not only labor productivity, but also which parts of the structure should be completed first, so as to plan ahead the coordination of scaffolding and location for temporary material storage among other constraints (Kim et al. 2014).

This type of inter-related queries can be, partially at least, supported by a BIM application if the right type of semantics is captured in the model, Following the example, the productivity is not only a function of the number of courses or masonry units installed per hour, but it is also a function of any internal feature the wall may have - such as openings, reinforcement or decorative patterns. An additional level of complexity arises when dealing with integration of different building systems. In this situation, productivity and cost are very much dependent on the degree of on-site coordination required among different trades involved. Given the increasing trend on prefabrication with deeper levels of systems integration, new strategies for masonry construction will be necessary. In particular, the computational challenge involved is not so much about the automatic generation of geometric detail, which in many cases may not be necessary, but the definition of the semantics of different masonry features that are relevant in the construction process.

In this research, we associate different wall features to different methods of abstractly decomposing masonry wall models. These abstract decompositions, called 'regions', denote domain-specific perspectives on the geometry of masonry walls, and are tied to one or more design requirements for which queries may be formulated. The concept of region is influenced by work on the ontology spatial boundaries (Smith and Varzi 2000) and on the ontology geographic objects (Smith and Mark 2003). In a pragmatic sense, the 'region' can be thought of as an arbitrary subdivision of a masonry wall into chunks with a special meaning for particular stakeholders. Implicitly, each chunk corresponds to an aggregation of masonry units and other components with associated domain-specific semantics.

For example, for a mason, the wall may be decomposed into sets of consecutive masonry courses (i.e. course regions), each with information regarding direction of assembly and boundary conditions that require special consideration. For a structural engineer the same masonry wall may be decomposed into different load-bearing regions for reinforcement design. Once defined, reinforcement regions may be used by mechanical engineers as volumetric references for the layout of piping and duct-work penetrations. These in turn define mechanical regions, to be used as reference for coordination with other domains and trades.

In this regard, special consideration has to be made to the identification of region boundaries. Examples of masonry features that constitute clear region boundaries include: control and expansion joints, relief angles, masonry recesses and inlays, and openings that decompose a wall into different, view-dependent geometric regions. In these cases the regions have a clear physical delimitation that coincides with that of the feature being represented (i.e. a *bona fide* boundary). In other cases however, the boundaries of the masonry feature is not self-evident, requiring in turn the arbitrary definition of its geometry (i.e. fiat boundaries). For example, a region representing a masonry feature such as a corner between two masonry walls may be delimited in different ways, depending on the perspective and requirements associated to the corner (see Figure 24, page 34). Other arbitrary region delimitations may be defined for different aspects of the wall lifecycle, such as sequence of erection, scaffolding and shoring requirements, productivity evaluation, condition assessment, etc. Because of this arbitrary delimitation, the definition of region boundaries by *fiat* needs to be a matter of agreement between stakeholders and domain experts. This particular aspect is discussed in more detail in section 3.3.2.3.

It is important to notice that masonry is fundamentally tri-dimensional, and therefore regions and region boundaries may be described not only along surface of walls, but also through the depth of walls (or other masonry structures). Such is the case for multi-wythe walls, cavity wall, or wall veneers with different backup systems. The principle of wall decomposition in depth allows the description of masonry layers under the same conceptual framework of in-plane regions. In this way it is possible to abstract away masonry layers from its constituting physical components such as flashing, ties, moisture barriers, insulation and the like. Such components can be either implicitly associated to its layer-region or explicitly propagated, thus supporting the generation of additional levels of detail on demand.

Figure 1 illustrates the concept of region, and how regions may be recursively decomposed into smaller sub-regions. By default, a wall region is defined as maximal, representing the whole wall. On the other extreme, a region may be minimal, that is, the smallest possible subdivision of the wall, which corresponds for practical purposes to the space occupied by half a brick or CMU block. In between both extremes, regions may be described at an intermediate level, as depicted in Figure 1. Regions can be implemented parametrically, and the specification of their parametric behavior may be specified by a constraint language such as the Building Object Behavior language (Lee et al. 2005; Cavieres et al. 2011).

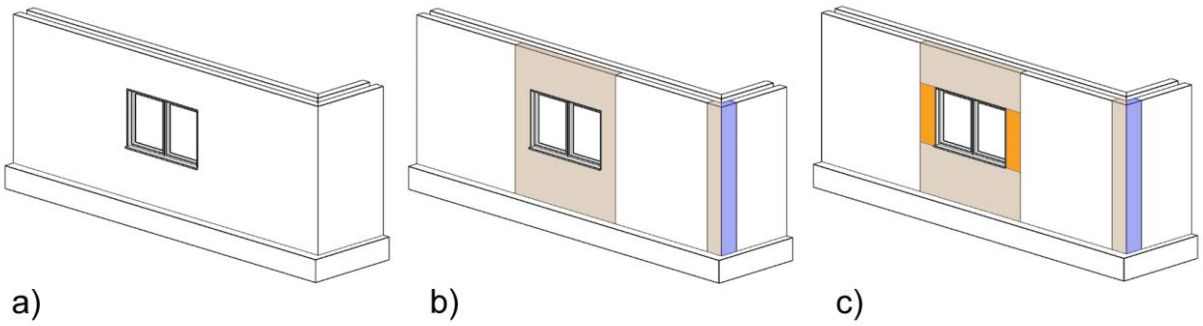

Figure 1 Region top-down decomposition according to the insertion of a window opening.

2.4 Regions and Levels of Development

In a design workflow certain activities can only happen if the right type of information is available for all stakeholders involved. This is especially important in a BIM enabled workflow, where queries and data exchanges require valid content from source models. In the context of relational databases, the outcome of predefined queries is called views (Elmasri 2006). A similar idea applies in the context of IFC, where standardized subsets of the source model (i.e. a model view definitions or MVD) are prescribed in order to support common data queries and exchanges between applications (Hietanen 2006; Venugopal et al. 2012).

Additionally, in order to ensure that the required levels of information for different design activities are present in the source model, the American Institute of Architects (AIA) has proposed the concept of Levels of Development (LOD) (AIA, 2013). The LOD concept has been adopted and extended by the BIM Forum, a consortium of organizations and companies representing the AEC industry, with a strong focus on developing and promoting best practices for virtual design and construction (VDC).

A LOD is specified according to a convention, ranging from LOD 100 to LOD 500. At LOD 100 the model is at a conceptual stage with few, coarse building elements. At LOD 500 the BIM model is fully detailed, with a geometric description of all construction elements needed for field verification of as-built information.

In practice, architect develops a level 100 model as part of conceptual design stage. As the design progresses into design development and construction documentation, the model is enriched with additional information and detail, typically up to LOD 300. On the other hand, general contractors and some sub-contractors are increasingly demanding BIM models with higher levels of development, normally at LOD 400.

In order to reduce the gap between the levels of information provided by the design model (LOD 200-300) and the construction model (LOD 400), the BIM Forum has proposed the addition of LOD 350. This new level was considered necessary for a BIM model to support more effectively interdisciplinary collaboration and trade coordination (Reinhardt and Bedrick 2013).

More recently, the BIM-M initiative has adopted the concept of LOD as reference for the specification of information requirements that need to be supported by masonry specific BIM applications. The use of LOD for masonry is described schematically here, as adaptation from the recent

report by the TMS BIM Committee (TMS 2014). This description is not intended to be definitive or complete, but should give the reader the basic idea of the LOD in BIM for masonry:

- **LOD 100:** Mid-plane wall surfaces and floor plates are modeled. No masonry 2D pattern. No wall openings.
- **LOD 200:** Walls defined through the thickness by layers. Walls checked for modularity with masonry rules.
- **LOD 300:** Wall thickness shown. Masonry 2D pattern mapped to walls. Layers can be further specified for structural masonry layer, insulation layers, and veneer layers.
- **LOD 350:** Vertical reinforcement and bond beams shown. Control and expansion joints are shown. Internal mechanical and electrical systems along with penetrations are shown at schematic level. Individual masonry units and other accessories are shown only when strictly necessary.
- **LOD 400**: Individual standard masonry units modeled. Location for masonry reinforcement represented.
- **LOD 500:** Condition of mortar modeled. Exact location of anchors and wall ties modeled. Geometric attributes of custom masonry objects present in model.

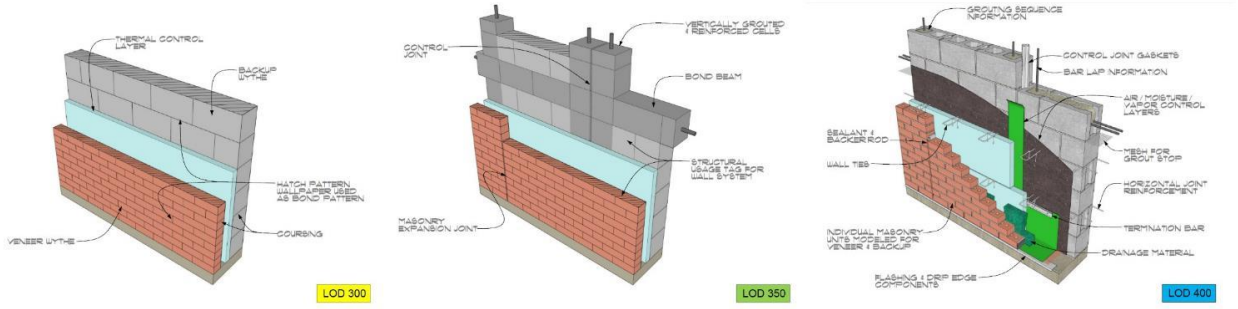

Figure 2 Evolution of LODs in Masonry. Source: TMS BIM-M COMMITTEE REPORT, January 2014.

The concept of masonry regions was developed by this research to facilitate the transitioning between levels of development. A special focus has been on supporting LOD 350, per industry consensus that this level provides the information necessary for interdisciplinary collaboration and trade coordination without the need for intensive geometric detailing.

In this regard, the concept of region is intended to enable the selective addition of geometric information wherever needed. For instance, more detail could be added, either manually or algorithmically, only to specific regions (see Figure 3 in page 9). Such an approach would be useful for the generation of virtual mock-ups and other high resolution models for performance and constructability analysis, without significant compromise in computational performance. Leaner BIM models are also important to facilitate change management, and to support design exploration and collaboration in masonry workflows.

3. RECOMMENDATIONS

In Report 1 of this project, submitted on 15 March 2015, a series of three main activities were identified as future steps for this research, which are being delivered here. Those were:

- The identification of relevant masonry wall region decompositions needed to satisfy current workflow requirements. This effort corresponds to the Deliverable 1 of the project (Specification of a Top-Down Modeling Approach) and it is initially addressed in general terms in the section 3.1 discussed below. Sections 3.2 and 3.3 complement section 3.1 by providing more specific details about the semantics of decomposed parts and software functionality.
- The formalization of the concept of region under a proper schema, in similar way to the one proposed by the Masonry Unit Database (MUD) project (see MUD final report dated 17 June 2015). Later on it became clear that the MUD schema should be extended in order to provide a comprehensive conceptual data model for masonry walls. This conceptual model includes all the major physical and abstract entities that comprise a masonry wall assembly, including masonry wall features, masonry wall components (e.g. Masonry units and accessories) and masonry wall regions. This effort led to the development of a conceptual model for masonry walls based on the semantics of IFC, which is described in section 3.3 Conceptual Model for Masonry Walls.
- The specification of general software functionality for BIM-M applications, including a preliminary formulation of geometric operators and parametric behavior for the creation and modification of masonry wall regions. Additionally, rules for semi-automatic and /or automatic generation of additional levels of geometric information according to pre-specified LODs need to be formulated in the future. The description of these efforts are part of the deliverable 3. Software Specification, which is addressed in the section titled “General Software Functional Specification”.

3.1 Top-down Modeling

The research proposes that formally defined region decompositions, particularly intermediate region decompositions, may provide the appropriate levels of abstraction for representing the semantics of masonry within acceptable levels of computational cost. Since a region is a view-dependent abstraction associated to specific design requirements, a masonry wall model may have many simultaneous and possibly overlapping regions. This is the case when a masonry wall may be subdivided into load-bearing regions for structural analysis, regions for thermal analysis and regions for planning sequences of erection, to name a few.

Therefore, for a BIM model to provide support for masonry specific queries, it must have knowledge not only of the type of design features a masonry wall may have, but also of the type of view-dependent regions a masonry wall may be decomposed into, as result of the addition of such features. It is the chained association between wall features, and between regions and domain-specific requirements that provide the semantics needed for the implementation of masonry models that can be queried effectively without the cost of too much geometric detail.

Since an intermediate region is a decomposition of a masonry wall into smaller chunks without reaching the level of resolution of individual masonry units, it can also be seen as an implicit, domain specific type of aggregation of masonry units and components. Figure 3 illustrated this idea. The overall

wall on the left is represented abstractly as a maximal region, which was decomposed into sub-regions based on the insertion of the opening for the window and the corner condition with the next wall. The process of decomposition follows the top-down modeling approach introduced in section 2.1, where designers typically starts with a simple model of the wall geometry, with some intent regarding the masonry material to be used.

The placement and dimensions of openings and other masonry features has to conform to the underlying modular grid of the maximal region. The spacing of the grid axis is a function of the nominal dimensions of the main masonry unit of the wall, thus supporting modular coordination all systems depending on the wall.

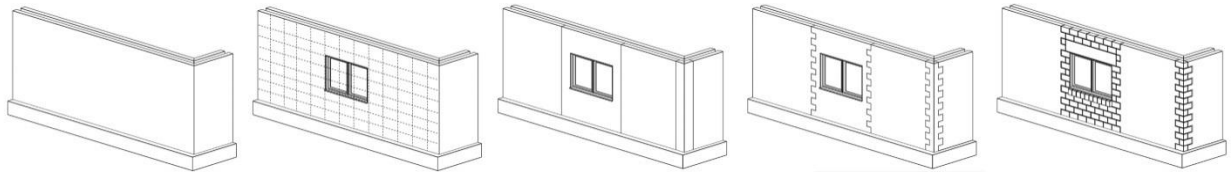

Figure 3 Sequence of insertion of a masonry opening feature. Top-down decomposition of regions also recognizes corner condition as a masonry feature. Additional LOD should be selectively added to specified regions.

As mentioned before, special performance or constructability requirements may need the addition of extra geometric information just on some portions of the wall. Such is the case around the opening and at the corner of the wall. These sub-regions may be described at LOD 400, which include the description of individual masonry units and other components such as ties anchors, flashing and insulation, etc. The rest of the sub-regions of the wall may be kept at LOD 200.

It is important to notice that the semantics of the decomposed masonry regions does not imply that the wall *has been physically subdivided* into discreet parts, but only abstractly subdivided into virtual boundaries similarly to how countries are subdivided into various regions based on recognizable geographic features. The exact shape and location of country boundaries are, to a large extend, arbitrary and subject to consensus. In the same token, the exact boundaries of a masonry feature (e.g. a corner) are not self-evident and therefore have to be agreed upon. For this reason a decomposed masonry wall needs to be considered as a single wall unit, with many possible and overlapping region decompositions.

In the following subsection some possible view-dependent are discussed.

3.2 View Dependent Region Decompositions

Special façade detailing, custom and cut units, structural load distributions, MEP clashes, scaffolding assembly sequences, cost estimation and condition assessment are some examples of possible views derived from the model as result of a query. In what follows five query scenarios are discussed, for which specialized regions need to be generated from the BIM model. Notice that these scenarios represent only a small number of possibilities identified by the BIM-M initiative. However, they are considered to be representative enough as to exemplify the theoretical approach adopted. Figure 4 provides an overview of the view-dependent region decompositions considered.

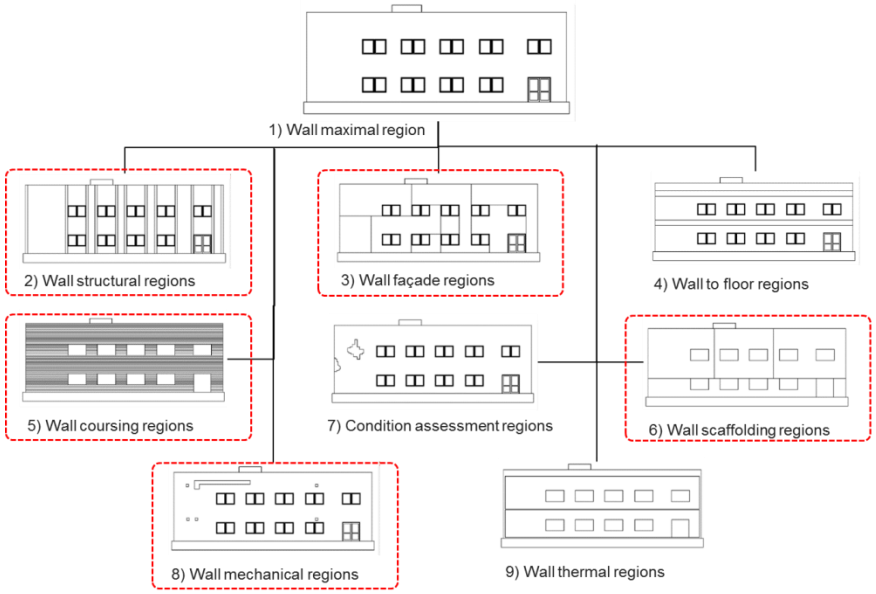

Figure 4 View dependent regions.

3.2.1 Regions for façade design and brickwork

In the façade design scenario, the primary system of interest is the "skin" of the building; the internal masonry construction is subsidiary to the appearance and performance of the skin. During early stages, the geometric model of the façade is lightweight, and probably represented using 2D surfaces or simple solids. In case of masonry being selected as the main material, the use of a "wall paper" representation may be enough to capture the intended patterning and texture of masonry at an LOD 100.

In case of complex brickwork, the maximal region corresponding to an entire wall may be decomposed into intermediate regions denoting different brickwork patterns, as well as inlays, recesses and corbels. Each region may have an individual bonding pattern (i.e. a hatch or "wall paper") object attached to it for rendering purposes, plus specific information on the properties of the unit types, bonding pattern, mortar and accessories used.

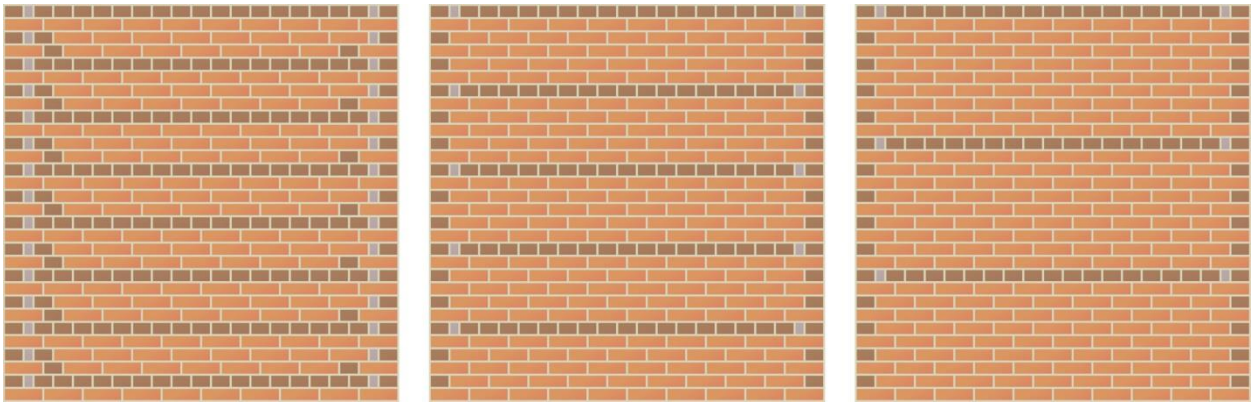

Figure 5 Brickwork patterning. Each sub-pattern may be depicted as brickwork region with specific information associated (e.g. color, texture, and manufacturer). Images by Jonathan Riley, available under Creative Commons license.

At later stages however, more detail is needed, especially regarding modular coordination of bonding patterns and coursing with individual features of the façade. In this case, overall dimensions need to be coordinated with such features, for example in situations like corners, openings, decorative elements or structural conditions (e.g. pilasters and expansion joints).

The primary use of this model is to support early decisions on the shape and overall appearance of the building skin, and the relationship between specific masonry unit types with different compositional patterns (see Figure 5A). Some initial concern about procurement, performance and constructability, maintenance and overall cost may arise as well. The output model describes regions of pattern differentiation that may be used for general and mason contractors for assembly planning and coordination.

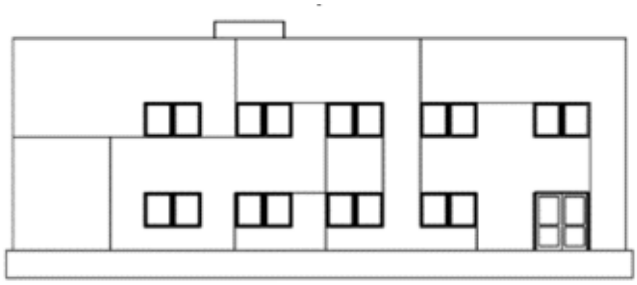

Figure 6 Figure 6 A schematic5A Schematic decomposition of an arbitrary masonry facade patterning, that is, an architectural point of view.

3.2.2 Regions for structural design

The structural model represents the load bearing masonry as a set of planar elements for subsequent input into different structural analysis programs, such as Bentley Ram Elements, which has been enhanced to work with structural masonry. Joints in the wall are properly modeled so that the conditions for lateral stiffness of the building are represented correctly. Openings in the load bearing walls are modeled. Non-load bearing walls are modeled so that their self-weight can be exported to structural analysis.

The generation of this information can be facilitated by the identification of load-bearing regions within the design model that correspond to the set of planar elements and plane boundaries for joints and openings. The output model describes regions of the wall where grouted cells, bond beams and other forms or reinforcement need to be considered by other stakeholder in the team.

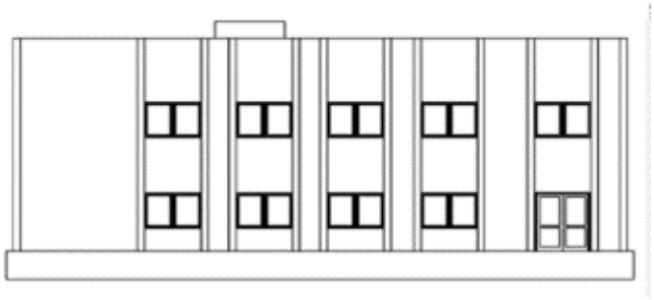

Figure 7 A schematic decomposition of a wall from a structural point of view.

3.2.3 Regions for mechanical systems

The structural clash detection model needs to represent only those masonry elements that contain reinforcement and grouted cells. In current clash detection applications, false clashes are identified between pipes and duct-work with unreinforced portions of walls. Such clashes are not relevant from the structural performance of masonry walls. Furthermore, it is part of mason's skill set to solve these clashes on site, by cutting CMU blocks and bricks as needed to resolve penetrations.

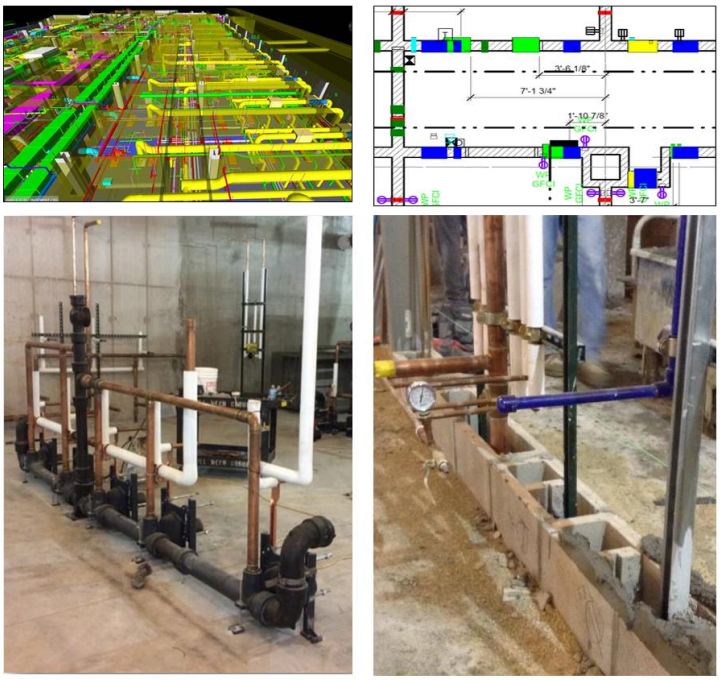

Figure 8 Example of prefabrication of piping systems deeply integrated with a CMU wall. Clash detection used for identification of safe regions for penetration sleeves. Images courtesy of Scott Conwell and the International Masonry Institute (http://www.imiweb.org).

However, masons require guidance when mechanical systems pass through bond beams or vertical grouted cells. Moreover, giving the increasing use of prefabricated piping and duct-work, it is recognized that better coordination among trades will be necessary (Conwell 2015). For this reason, geometric intersections between masonry walls with other systems can be represented by localized regions of the wall, conveying the necessary information to all the sub-contractors involved. Figure 8 illustrates an example of a prefabricated pipe assembly integrated inside CMU masonry walls. It also shows drawing conveying regions for penetration sleeves derived from the BIM model. Figure 9 shows the schematic regions of masonry walls where potential clashes have been highlighted.

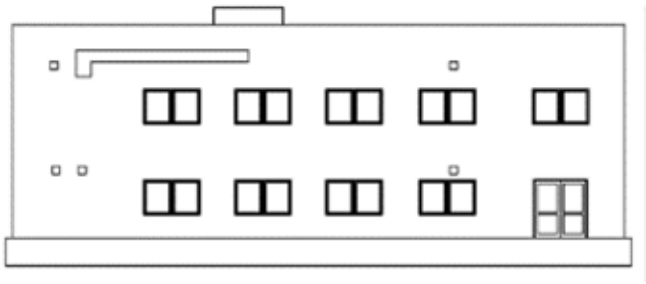

Figure 9 Schematic regions for clashing with mechanical systems.

3.2.4 Regions for quantity take-offs

In quantity take-off, regions of the masonry walls are tagged when similar components and construction processes are anticipated. In this way associated quantities can be aggregated for cost estimating calculations. Special conditions that require higher levels of reinforcement, large number of cuts, special

units, or unusual geometry are considered separately as wall production rates in these regions are likely to be lower. The output model of this task consists of regions to be used next for cost estimation and construction planning activities.

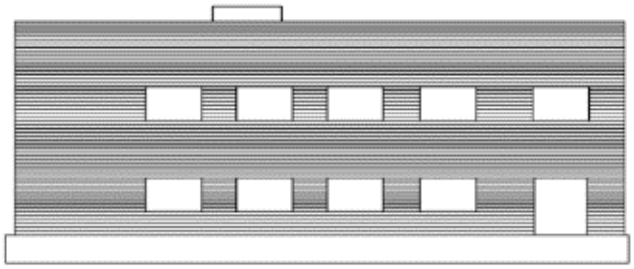

Figure 10 Schematic regions for quantity take-off based on masonry courses.

3.2.5 Regions for cost estimation and construction planning

The cost estimation and construction planning scenario injects the idea of time into the BIM-M model. The queries envisioned are of the sort that might be expected in the pre-construction stage as well as during construction to solve emerging issues.

Such queries may be intended to identify the temporary location of material at the construction site (i.e. staging), what components have already been installed, what are the next sequences of construction activity, or other questions related to productivity. The level of information in this model may vary depending on the complexity of the assembly. Thus, regions with conventional configuration may be defined starting at LOD 300, while more complex regions may need to be generated as virtual mock-ups at LOD 500. All these region models will have to be continuously updated during the construction process to reflect as-built information. This means that the wall region schema will be required to be subdivided into sub-regions that represent units of production (e.g. the geometric regions depicting the production of one masonry crew for one week). In addition, the BIM-M model will need to refer to additional objects that are not intrinsic to the masonry wall, such as material staging, shoring and scaffolding.

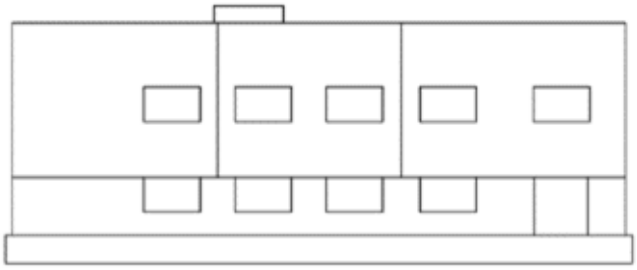

Figure 11 A schematic view of possible regions representing coverage area for scaffolding.

3.3 Conceptual Data Model for Masonry Walls

The implementation of masonry specific BIM applications entails first and foremost the definition of conceptual data models (i.e. schemas) that allow a standard representation of masonry entities and relationships in a machine-readable format. This issue has been initially addressed by the BIMM initiative with the development of the MUD project, which focused primarily on a database schema for masonry units. The next step in MUD is to include within the same framework the definition of masonry components and accessories that are most commonly used in masonry walls.

However, the representation of masonry walls requires more than the definition of individual components and accessories that can be purchased off the shelf. In fact masonry walls are characterized by a series of intermediate assembly features that result from particular arrangements of masonry units, components and accessories. Features such as wythes, veneers, corbels, recesses, corners and bond beams are some examples of features that are intrinsic to masonry walls for which currently there is no schema definition. During the development of the Masonry Wall Definition (MWD) this issue was recognized as a priority for the subsequent specification of BIM-M software applications.

Therefore this report proposes an integrated conceptualization for all masonry entities under a common schema framework. In order ensure semantic compatibility and extensibility with current BIM applications the proposed schema has been developed using IFC as main reference. The primary purposes however is not to enable interoperability of masonry objects for which a standard schema does not yet exist. On the contrary, the intent is to start building such a standard schema using a well-established conceptual model as a foundation. We hope this decision will facilitate discussion and collaboration with various actors in the AEC industry, including software vendor.

Since the semantics of masonry entities are being defined using IFC as reference, a brief overview of main IFC classes is presented first (IFC 2x3). The core inheritance structure of IFC is introduced in order to describe the properties and relationships that are reused for the definition of masonry wall entities. Each of these is elaborated in detail afterwards, providing a foundation for the specification of BIM-M applications described in section 4.

Note: The modeling language used in this report to describe the relation between IFC and the proposed masonry wall schema is an UML dialect called SysML. This language has been used previously in the BIM-M Benchmark project for diagrammatic purposes. In this project we decided to use this language because of a series of documentation and traceability resources available, and for the quick generation of XML schemas for prototyping. Some original EXPRESS-G diagrams are reproduced from buildingSMART official website.

3.3.1 IFC Definitions

The Industry Foundation Classes (IFC) is a standard data schema (data model) developed and maintained by buildingSMART International, which registered it as an official ISO standard ISO16739. The IFC schema is the basis for a set of neutral, open file formats to support data interoperability among Building Information Modeling applications.

The IFC schema is essentially an entity-relationship (ER) model defined using the EXPRESS language. It is organized according to an object-based inheritance structure based on a series of abstract low level constructs. The construct IfcRoot defines the most general entity, from which most of IFC entities

inherited from. These entities are called rooted entities. Among the most important inherited properties of a rooted entity is the Globally Unique Identifier (GUID), a name, a textual description and revision control attribute. Non-rooted entities in turn do not have identifiers associated with them and can only exist at the instance level by direct or indirect reference to a rooted instance.

The IfcRoot has three main subtypes:

- IfcObjectDefinition: describes tangible object occurrences and types.
- IfcRelationship: describes relationships among objects.
- IfcPropertyDefinition: describes dynamically extensible properties about object.

3.3.1.1 IfcObjectDefintion

IfcObjectDefinition is the "generalization of any semantically treated thing or process, either being a type or an occurrence". For that purpose IfcObjectDefinition is divided into two main categories: IfcObject, which provides core definitions for object occurrences such as products, processes and resources, and IfcTypeObject which allows extensibility of IfcObjects core semantics. Objects defined under IfcObject can belong to one of the following categories.

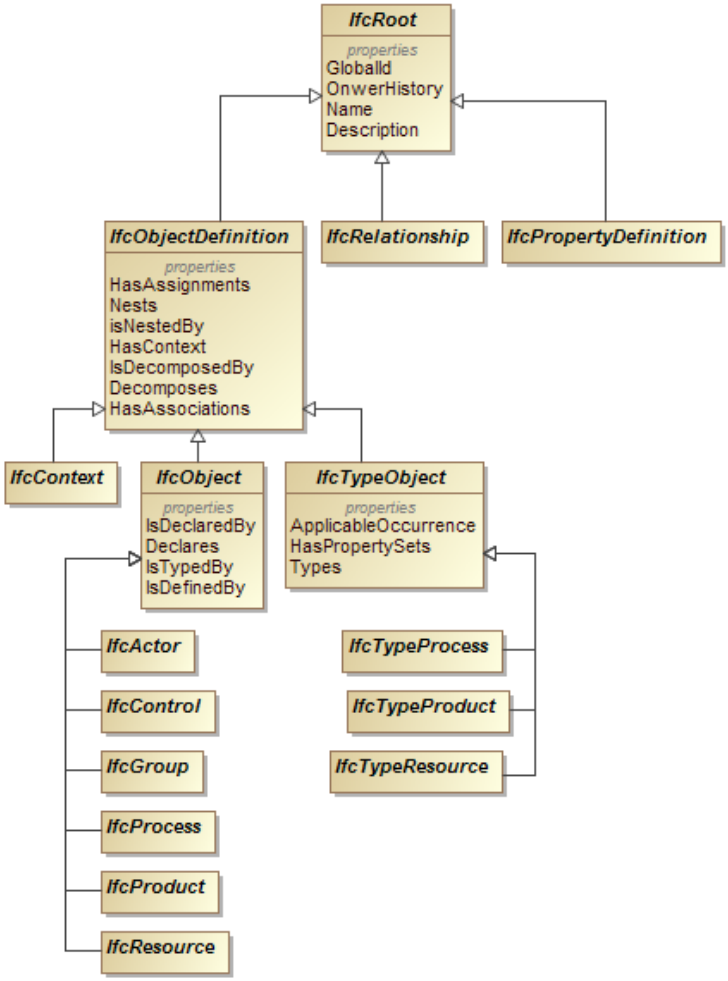

Figure 12 Basic IFC definitions, using SysML block diagrams.

- **IfcActor**: for the representation of stakeholders, either individuals or organization.
- **IfcControl**: for the representation of rules for the control of time, cost, scope and work orders.
- **IfcGroup**: for the representation of collections of objects with a special purpose.
- **IfcProduct**: for the representation of building entities such as site, space, wall, etc.
- **IfcProcess**: for the representation of time-dependent concepts such as tasks, events, sequences, etc.
- **IfcResource**: for the representation of entities subject to usage and limited availability, such labor and equipment.

3.3.1.2 IfcRelationship

IfcRelationship is the most general class of objectified relationships between objects in IFC. It defines five fundamental relationship types that can be further specialized:

- **IfcRelDecomposes**: A *part-of* relationship, such as windows being part of walls.
- **IfcRelAssigns**: A assignment relationship indicating allocation of resources.
- **IfcRelConnects**: A connectivity relationship between objects, such as between a beam and a wall.
- **IfcRelAssociates**: An association of objects to external references, such as product libraries.
- **IfcRelDefines**: A general type-definition relationship, where object instances can be typed dynamically to particular types

3.3.1.3 IfcPropertyDefinition

IfcPropertyDefinition provides the generalization of all characteristics that may be assigned to objects. Through this mechanism common object information about objects can be shared among all their subtypes and instances. IfcPropertyDefinition also provides the means for dynamically attaching new sets of properties to object occurrences.

3.3.1.4 IfcProduct

IfcProduct is the general class for any object that can be described geometrically. Subclasses of IfcProduct usually have a shape representation and an object placement associated within them. IfcProduct includes all objects that are manufactured, supplied or created on-site as direct or indirect result of the construction process. These objects are referred as elements and subtyped under IfcElement whenever they have a geometric nature. The definition of IfcProduct also allows the description of non-physical objects such as space objects, which are created indirectly by the boundaries of physical elements. Spatial objects are subtyped under IfcSpatialElement. Finally, an IfcProduct can be designated for permanent or temporary use, such as scaffolding and formwork, having a possible relationship with instances of IfcProcess by means of IfcRelAssignsToProcess and IfcResource by means of IfcRelAssignsToResource.

3.3.1.5 IfcElement

IfcElement generalizes all components that make up a building, including physical objects as well as void elements such as holes. Elements can be permanent or temporary such as framework and scaffolds. Typically, elements are prefabricated and assembled on-site or completely built on-site. The aggregation of different elements into assemblies is mediated through the use the objectified relationship IfcRelAggregates which specifies a non-ordered aggregation relationship, as opposed to IfcRelNests.

Elements can have material properties and quantity information assigned to them through IfcRelAssociatedMaterial and IfcRelDefinedByProperties. An element can also be declared as an occurrence of a specific object type by means of IfcRelDefinesByType. This mechanism allows predefined sets of domain-specific properties to be added to the element occurrence, thus allowing the extensibility of the core semantics originally defined for the element. Figure 13 illustrates this mechanism.

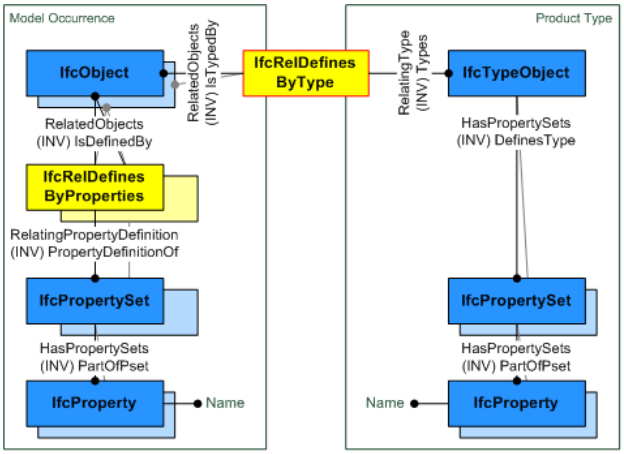

Figure 13 Assignment of a predefined product type to an object occurrence using IfcRelDefinedByType objectified relationship.

Elements can also be grouped together according to specific perspectives. For example, elements that participate in the satisfaction of given function may be grouped using an instance of IfcGroup with the inherited attribute *ObjectType* = "ElementGroupByFunction".

Quantities related to an element can be defined using IfcElementQuantity and attached to the element using the objectified relationship IfcRelDefinesByProperties.

Finally, multiple different geometric representations can be associated to an IfcElement by means of the IfcProductDefinitionShape. The geometric representation of an element is dependent of its location, specified by IfcLocalPlacement. The list of core properties defined by IfcElement (as inverse attributes) is provided in Figure 14.

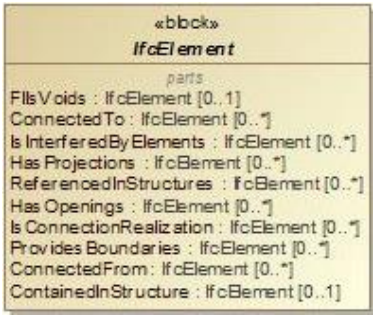

Figure 14 Inverse attributes defined for IfcElement and its subclasses.

3.3.1.6 IfcBuildingElement

IfcBuildingElement comprises all major elements of a building that perform a function and are result of the construction process. These include foundations, columns, walls, roofs, doors and windows, etc.

Because of the special relevance of this class for the formal definition of masonry walls, the set of core relational properties associated to IfcBuildingElement are described here based on the original IFC documentation. The potential applicability of each relationship in the context of masonry walls is exemplified with a list of possible use cases under the description of each IfcBuildingElementProperty:

a) **Grouping** - being part of a logical group of objects
   - objectified relationship: IfcRelAssignsToGroup
   - object referenced by relationship: IfcGroup (and subtypes)
   - inverse attribute: *HasAssignment*

   Masonry use-case:

   - Masonry walls can be grouped according to different construction sequences.
   - Masonry walls can be grouped according to complexity and cost.
   - Masonry walls can be grouped by state of completion.
   - Masonry walls can be grouped by different functions.
   - Masonry wall can be grouped by same condition assessment (FM view).
   - Masonry units and components can be grouped within individual masonry walls according to different stakeholders' perspectives. Each group can be represented geometrically by a region. The ability to provide a geometric representation for various forms of grouping that are important in the context of masonry construction is one of the main motivations behind the formulation of the concept of masonry regions.

b) **Work processes** - reference to work tasks, in which this building element is used
   - objectified relationship: IfcRelAssignsToProcess
   - object referenced by relationship: IfcProcess (and subtypes)
   - inverse attribute: *HasAssignments*

   Masonry use-case:

   - Masonry walls can be assigned to different construction processes.
   - Groups of masonry units and components can be assigned to different erection or assembly sequences. These groups can be represented geometrically as erection or assembly sequence regions.

c) **Structural member reference** - information whether the building element is represented in a structural analysis model by a structural member
   - objectified relationship: IfcRelAssignsToProduct
   - object referenced by relationship: IfcStructuralMember (and by default IfcStructuralCurveMember)
   - inverse attribute: *HasAssignments*

Masonry use-case:

- Masonry walls can be represented by masonry specific IFC structural members.
- Masonry wall features with different structural behaviors can be represented by different structural regions, each with different structural properties associated.

d) **Aggregation** - aggregated together with other elements to form an aggregate
   - objectified relationship: IfcRelAggregates
   - object referenced by relationship: IfcElement (and subtypes)
   - inverse attribute (for container): *IsDecomposedBy*
   - inverse attribute (for contained parts): *Decomposes*

   Masonry use-case:

   - Building floors are decomposed into masonry walls (among other things).
   - Masonry walls are decomposed into masonry wall features (e.g. wall layers).
   - Masonry wall features are decomposed into masonry components and ingredients (explained later). Masonry features are represented geometrically by feature regions. Each region representing a masonry feature may have a different level of geometric detail associated to it. Thus, complex features that require constructability assessment or trade coordination may be modeled with more geometric detail than others. In any case the aggregation relations should remain the same (e.g. part count). The same applies to other relations discussed here.

e) **Material** - assignment of material used by this building element
   - objectified relationship: IfcRelAssociatesMaterial
   - object referenced by relationship: IfcMaterialSelect (and selected items)
   - inverse attribute: *HasAssociations*

   Masonry use-case:

   - Assignment of material to masonry is done at the component level and ingredient levels. Bricks are assigned certain types of clay, CMU block to certain types of concrete; rebars to certain types of steel, mortar and grout are assigned to certain types of cement mixtures and aggregates.

f) **Classification** - assigned reference to an external classification
   - objectified relationship: IfcRelAssociatesClassification
   - object referenced by relationship: *IfcClassificationNotationSelect* (and selected items, defaultIfcClassificationReference)
   - inverse attribute: *HasAssociations*

Masonry use-case:

- Masonry wall systems, features, components and ingredients can be assigned to standard classifications systems (e.g. MUD).

g) **Library** - assigned reference to an external library item reference
   - objectified relationship: IfcRelAssociatesClassification
   - object referenced by relationship: IfcLibrarySelect (and selected items, defaultIfcLibraryReference)
   - inverse attribute: *HasAssociations*

   Masonry use-case:

   - Masonry components and ingredients can be assigned to manufacturer or provider specific libraries.

h) **Documentation** - assigned reference to an external documentation
   - objectified relationship: *IfcRelAssociatesDocumentation*
   - object referenced by relationship: IfcDocumentSelect (and selected items, defaultIfcDocumentReference)
   - inverse attribute: *HasAssociations*

   Masonry use-case:

   - Masonry systems, features, components and ingredients can be assigned to different stakeholders' documentation. For example, construction time, skill level, resources required and cost associated to each masonry system or feature may be part of the general / masonry contractor knowledge-base.

i) **Type** - reference to the common product type information for the element occurrence
   - objectified relationship: IfcRelDefinesByType
   - object referenced by relationship: IfcBuildingElementType (and subtypes)
   - inverse attribute: *IsTypedBy*

   Masonry use-case:

   - Occurrences of masonry wall systems can be specified using *ITypeBy* relationship. For example, cavity walls, multi wythe wall, stack walls can be defined with their own specific property sets and attached to generic masonry wall occurrences.
   - Occurrences of major masonry wall parts, such as insulation and barrier layers as well as different forms of masonry covering can be typed by this relationship.
   - Occurrences of masonry regions can also be typed by predefined, view dependent regions.

j) **Properties** - reference to all attached properties, including quantities
   - objectified relationship: IfcRelDefinesByProperties

- object referenced by relationship: IfcPropertySetDefinition (default IfcPropertySet)
- inverse attribute: *IsDefinedBy*

Masonry use-case:

- Occurrences of masonry wall features and regions can be further qualified using *IsDefinedBy* relationship. For example, different constructability metrics to could be added to features such inlays, recesses, quoins, corbels.

k) **Connection** - connectivity to other elements, including the definition of the joint
- objectified relationship: IfcRelConnectsElements
- object referenced by relationship: IfcElement
- inverse attribute: *ConnectedTo*
- inverse attribute: *ConnectedFrom*

Masonry use-case:

- This relationship allows the description of connectivity of masonry walls with other building systems, including other walls, which often require more specificity (see Realization below).
- This relationship also allows the description of connectivity between internal features and parts of a masonry wall. For instance, when certain layers are connected.

l) **Realization** - information, whether the building element is used to realize a connection (e.g. as a weld in a connection between two members)
- objectified relationship: IfcRelConnectsWithRealizingElements
- object referenced by relationship: IfcElement
- inverse attribute: *IsConnectionRealization*

Masonry use-case:

- The realization relationship applies to internal masonry features realizing the connection between their 'container' walls to other building systems. For example, masonry recesses, niches or corbels can be used to receive steel joists. In this way, these features *realize* the connection. Since these features share in common a load-bearing functionality, they can also be grouped together using IfcRelAssignsToGroup relationship explained previously.
- The realization relationship applies to masonry features realizing the connection between their 'container' walls to other walls (masonry or otherwise) specially in corner conditions. In this way it is possible to identify wall corners as unique, distinct features, with their own set of properties associated to construction and structural performance. For instance, a corner can be resolved in many different ways, each implying a different realization relation. Thus, two masonry walls can be connected by interlocked masonry units, quoins, by an isolation joint, by steel studs or concrete columns, etc.
- The realization relation also applies to masonry features realizing the connection between other features and parts of the same masonry wall. For example, mortar joints

realizing the bonding between masonry units. Since different types of bonding mechanisms can be used, the connection can be realized in different ways (e.g. in stack walls masonry units are connected directly one against each other). A brick veneer can be connected to a backup system in different ways as well, each one implying a different realization relation.

-

m) **Spatial containment** - hierarchical assignment to the right level within the spatial structure
   - objectified relationship: IfcRelContainedInSpatialStructure
   - object referenced by relationship: IfcSpatialStructureElement
   - inverse attribute: *ContainedInStructure*

Masonry use-case:

- Since a masonry wall is proposed here as subtype of IfcWall, it can only be assigned hierarchically to one spatial structure by means of IfcRelContainedInStructure relationship.

n) **Spatial references** - nonhierarchical reference to one or more elements within the spatial structure (e.g. a curtain wall, being contained in the building, references several stories)
   - objectified relationship: IfcRelReferencedInSpatialStructure
   - object referenced by relationship: IfcSpatialElement
   - inverse attribute: *ReferencedInStructure*

Masonry use-case:

- Since a masonry wall is proposed here as subtype of IfcWall, it can be referenced non-hierarchically by many spatial structures using IfcRelReferencedInStructure relationship.

o) **Boundaries** - provision of space boundaries by this building element
   - objectified relationship: IfcRelSpaceBoundary
   - object referenced by relationship: IfcSpace
   - inverse attribute: *ProvidesBoundaries*

Masonry use-case:

- Since a masonry wall is proposed here as subtype of IfcWall, it can provide the boundaries for instances of IfcSpace.

p) **Coverings** - assignment of covering elements to this building element (note: normally covering elements are assigned to the space, only used for special cases)
   - objectified relationship: IfcRelCoversBldgElements
   - object referenced by relationship: IfcCovering
   - inverse attribute: *HasCoverings*

Masonry use-case:

- Covering need to be assigned to spaces. Masonry walls can have material specifications assigned to layer features.

q) **Voids** - information, whether the building element includes openings, recesses or other voids
- objectified relationship: IfcRelVoidsElement
- object referenced by relationship: IfcFeatureElementSubtraction (default IfcOpeningElement)
- inverse attribute: *HasOpenings*

Masonry use-case:
- Since a masonry wall is proposed here as semantically equivalent to a subtype of IfcWall, it can also have voids for true openings, internal openings, niches and recesses.
- Voids can also apply for masonry wall features and other parts.

r) **Projections** - information, whether the building element has projections (such as a fascia)
- objectified relationship: IfcRelProjectsElement
- object referenced by relationship: IfcFeatureElementAddition (default IfcProjectionElement)
- inverse attribute: *HasProjections*

Masonry use-case:
- Since a masonry wall is proposed here as semantically equivalent to a subtype of IfcWall, it can have projections associated to it. For example, for corbels, pilasters and other protruded features.

s) **Fillings** - information whether the building element is used to fill openings
- objectified relationship: IfcRelFillsElement
- object referenced by relationship: IfcOpeningElement
- inverse attribute: *FillsVoids*

Masonry use-case:
- Since a masonry wall is proposed here as semantically equivalent to a subtype of IfcWall, it can have fillings associated to its voids. Bond beams and grouted cells, ore some examples.
- Fillings can also apply for masonry wall features and other parts.

3.3.1.7 IfcWall

IfcWall is the fundamental entity in IFC for the representation of all kinds of walls. An IfcWall represents a vertical or nearly vertical element that bounds or subdivides spaces. An IfcWall may or may not be designed to carry structural loads. An IfcWall may have wall openings for windows and doors, as well as niches and recesses. These are described by an IfcOpeningElement attached to the wall using the objectified relationship IfcRelVoidsElement (and the inverse property HasOpenings), which are inherited from IfcElement explained above. Figure 9 illustrates this relationship.

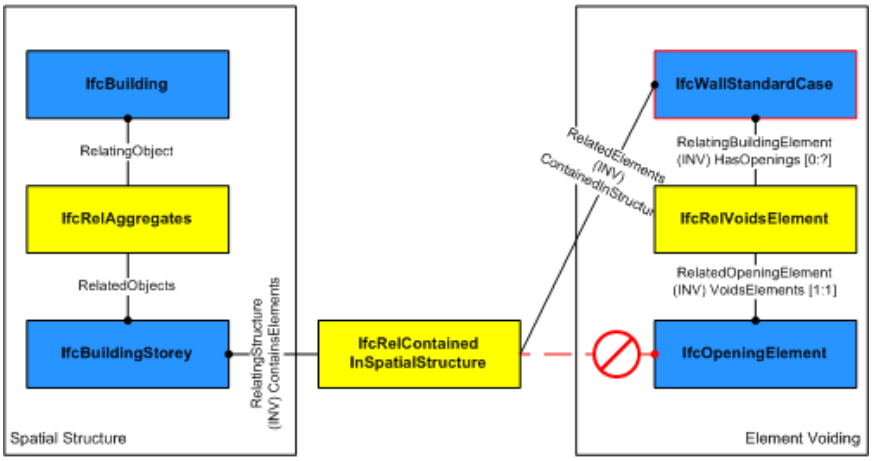

Figure 15 Relation between a wall and an opening in IFC through the objectified relationship IfcRelVoidsElement.

IfcWall has two subclasses:

- IfcWallStandardCase: Used for all occurrences of walls with a constant thickness along the wall path. The internal structure of these walls can always be described as a collection of layers, each one with different material properties, thickness and function (e.g. IfcMaterialLayerSet). The geometric representation of these walls is typically handled as swept solid along a main axis.

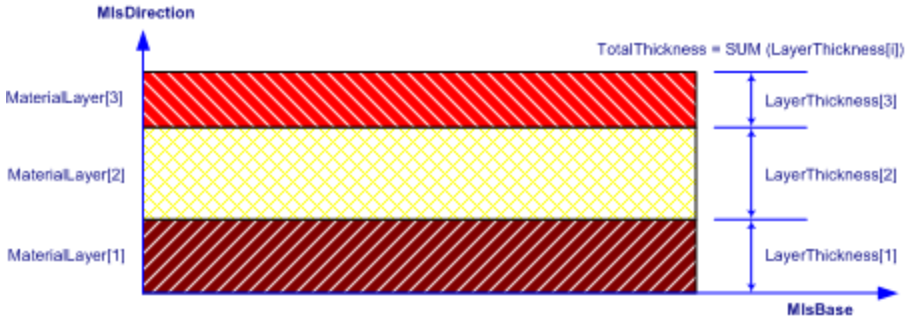

Figure 16 Structure of Material Layer Set.

- IfcWallElementedCase: Used for occurrences of walls that need to be represented explicitly as assemblies of subordinate elements. The decomposition of the assembly into parts gets established by the use of the IfcRelAggregates relationship.

All other occurrences of a wall remain defined under the more general level IfcWall. These include those with variable thickness along the path, polygonal footprints, non-rectangular cross sections (e.g. L-shaped retaining walls) and non-vertical walls. All IfcWall occurrences have base quantities associated to them, which include:

- Length (nominal)

- Width (nominal)
- Height (nominal)
- Gross Footprint Area
- Net Footprint Area
- Gross Side Area
- Net Side Area
- Gross Volume
- Net Volume
- Gross Weight
- Net Weight

Additionally, there is a series of wall specific property sets that can be attached to wall occurrences by means of IfcRelDefinesByProperties relationship. These include properties related to acoustic insulation rating, fire rating, thermal transmittance, load bearing capability, etc. For more details see table 185 in the IfcWall specification.

Figure 17 (page 28) illustrates the inheritance structure from IfcProduct down to the two subclasses of IfcWall. The diagram shows the list of properties belonging to IfcWall, including the properties inherited all the way from IfcRoot (e.g. GlobalD, Name, OwnerHistory and Description). These represent the core list of properties used in this proposal to define masonry walls, in order to maintain the semantic equivalence with original IFC class definitions.

3.3.1.8 IfcBuildingElementProxy

Most of building elements defined under the class IfcBuildingElement correspond to physical artifacts with a predefined meaning in terms of shape and functionality. Thus, the class IfcBuildingElement covers the description of common elements such as walls, slabs, beams, doors, etc. Sometimes however these categories do not cover all possible entities that might be relevant in the context of design.

For this purpose IFC provides an abstraction called IfcBuildingElementProxy. IfcBuildingElementProxy contains the same set of properties and relationships inherited by all subtypes of IfcBuildingElement, but without a predefined shape or function associated to it. According to buildingSMART, the purpose of IfcBuildingElementProxy is to work as **spatial place-holders** that later maybe replaced by special types of elements. The following use cases for IfcBuildingElementProxy are provided:

- The IfcBuildingElementProxy can be used to represent a particular volume of space needed to allocate some engineering function. That volume of space may later be assigned as a void within a larger building element, such as an opening, niche of recess.
- The IfcBuildingElementProxy can be used to exchange special types of building elements for which there is no current standard specification of semantics, either at the exchange level or at the application level.

As any subtype of IfcBuildingElement, an occurrence of IfcBuildingElementProxy stands to a number of inherited properties and relationships, such as object placement, shape representation, spatial containment, element composition, material associations (just one), object typing and predefined property sets.

The purpose and characteristics of IfcBuildingElementProxy make it an appropriate category for the definition of masonry wall regions under the IFC schema, as explained later in section 3.3.2.

3.3.1.9 Preliminary Discussion

The analysis of representational requirements for masonry walls, especially from the perspective of design workflows and Levels of Development (LOD) currently needed by the industry indicate that the semantics provided at the level of IfcWall are not enough. These semantics correspond roughly to what is already available today for the representation of masonry walls in most BIM commercial applications.

For example, in Autodesk Revit, a conventional approach for the representation of masonry walls is the use stack wall families with material layer sets. In fact, this approach is general enough as to represent any kind of multilayer walls, without any specific rules or constraints regarding masonry construction. Thus, the resulting stack wall model in Revit is semantically equivalent to **IfcWallStandardCase** previously described in section 3.3.1.7. Given the LOD requirements for masonry walls recommended by the TMS BIM-M Committee Report (January 2014), such an approach seem only appropriate to cover LODs 100 – 200.

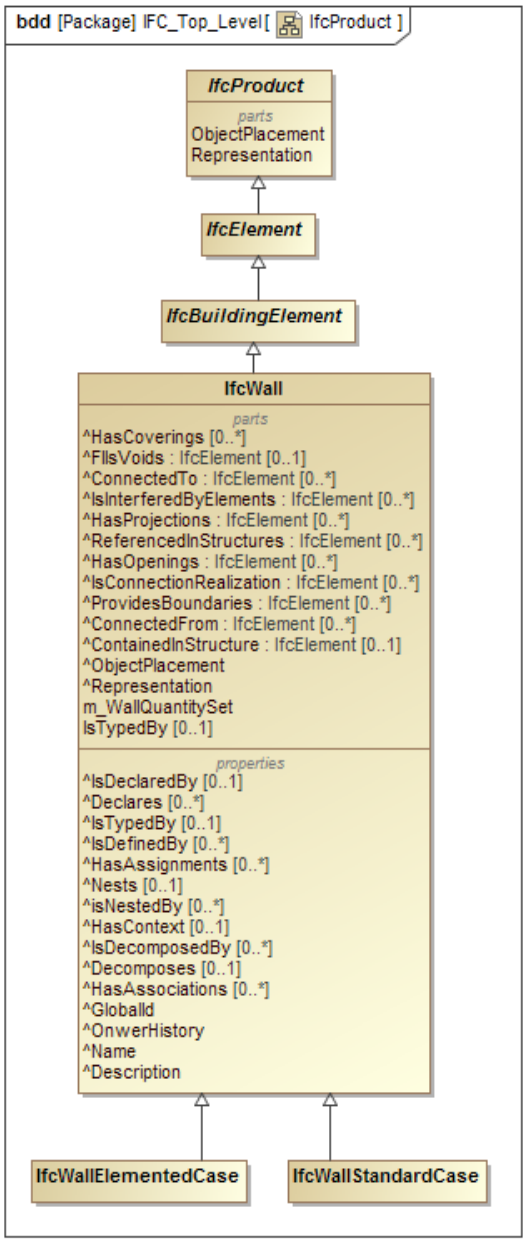

Figure 17 Inheritance structure for IfcWall, IfcWallStandardCase and IfcWallElementedCase.

The representation of masonry walls at LOD 300 may be accomplished within Autodesk Revit using a relatively new functionality called 'parts' (Revit 2014). Thus, a multilayer wall represented as a single solid with embedded material layer sets (equivalent to IfcWallStandardCase) can be converted automatically into series of geometrically discreet layers. Each layer object in turn can be further decomposed into smaller subparts, given a number of predefined geometric operators[1]. The resulting model would be semantically equivalent to an **IfcWallElementedCase** occurrence. This second characterization of a masonry wall as an assembly of different element parts (i.e. IFC elements) seems more appropriate to reach the semantics required for LOD 300 and above (Figure 18).

Because of the industry need to cover LODs higher than 200, a masonry specific new wall type is proposed in this report. This masonry-specific wall type is rooted on the semantics of IfcWallElementCase.

This section introduces the rationale for the newly proposed type. In order to explain the relationship between the proposed masonry wall type with the semantics of IFC, the section starts with low level definitions for masonry parts (e.g. masonry units and components). This is followed by the description of a new level of intermediate aggregation dedicated to the description of masonry features. Masonry regions are then introduced as abstract representation of masonry features. Finally the entire masonry wall assembly type is introduced at the highest level of aggregation.

The intent is to provide a comprehensive, unified schema for the representation of masonry walls. However, this is just an initial conceptual framework. It is by no means exhaustive or final. Significant development effort will be needed from all industry members to revise, modify and extend the preliminary schema and software functionality suggested here.

A discussion will be provided in section 3.4 Software Functionality regarding the geometric operators needed to build and maintain the geometric representation of masonry walls, features and regions according to the masonry schema. As mentioned in the case of Revit wall parts, much of this

functionality is already available in mainstream BIM applications, but they need to be tied to more refined and extended to cover with more domain-specific requirements.

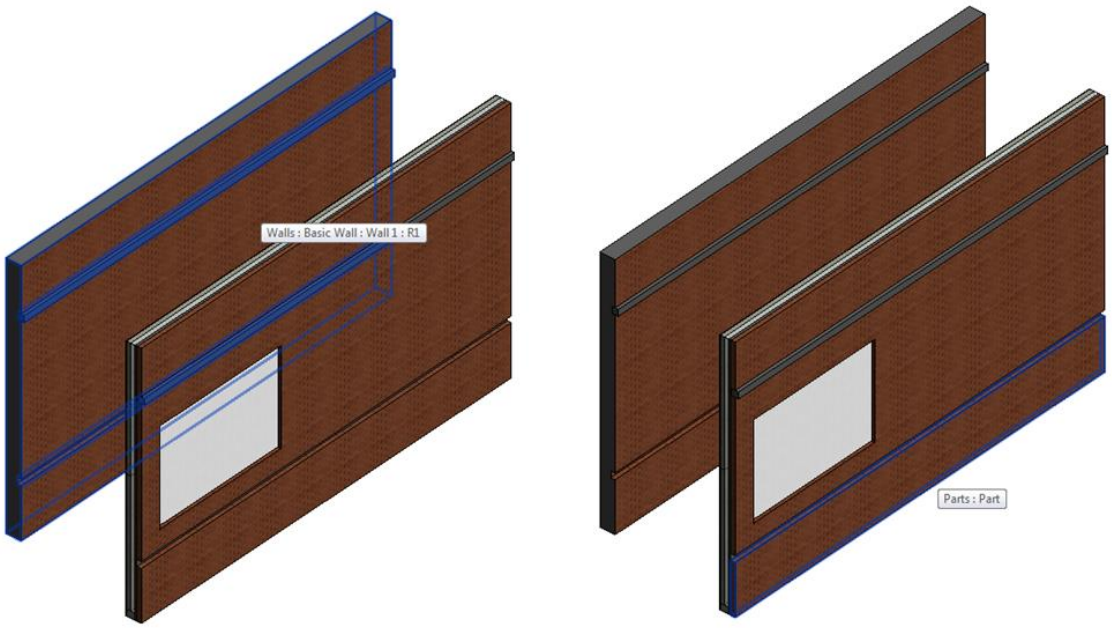

Figure 18 Models of masonry cavity wall in Revit 2015. In the left wall, all layers and masonry features are implicit in the model (e.g. material layer set) created as a Revit stacked wall. This representation is similar to IfcWallStandardCase . On the right, all layers and features are represented explicitly as independent parts of the wall, which is equivalent to IfcWallElementedCase.

3.3.2 Masonry Wall Definition Schema

Based on the observation made on the previous discussion, the proposed class of masonry walls is defined as semantically equivalent to a subtype of IfcWallElementedCase. Masonry walls can still be created within the characteristics of IfcWallStandardCase (i.e. swept solids with constant thickness and material layer sets) , but this option should be used only during early stages of design or when low levels of development such as LOD 100 and 200 are sufficient to cover the information requirements at hand.

When higher levels of development are needed, masonry walls should be modeled in ways equivalent to IfcWallElementedCase. For that purpose the research proposes the class **mwd_MasonryWall**[2] as semantically equivalent to subtype of IfcWallElementedCase (Figure 19).

[2] All masonry specific entities defined within the proposed schema have the prefix mwd_, which stands for Masonry Wall Definition. These new entities are represented in the subsequent diagrams as green boxes, to differentiate them from IFC entities.

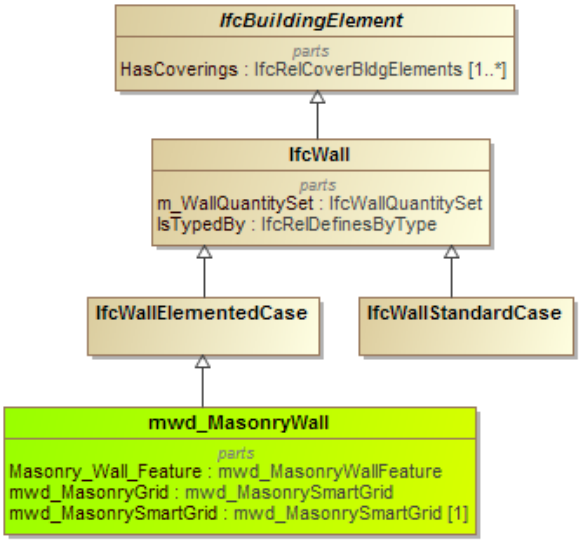

Figure 19 Masonry walls as subtype of elemented walls/

In order to provide a comprehensive and unified schema for masonry walls (i.e. mwd_MasonryWall), it is necessary to start defining schema definition for the parts and components that constitute a masonry wall assembly as well. Therefore, all parts of a masonry wall, including masonry units, rebar, joint wires, grout meshes, anchors ties, etc. are defined as semantically equivalent to subtypes of fcElementComponent. This allows the inheritance of a series of common properties that are relevant for all masonry parts and for which IFC already provides some general level definitions. The proposed masonry wall schema differentiates masonry parts into three main categories:

- mwd_MasonryComponent (section 3.3.2.1)
- mwd_MasonryIngredient (section 3.3.2.2)
- mwd_MasonryWallFeature (section 3.3.2.3)

A masonry wall feature represents an intermediate aggregation of components and ingredients (i.e. subassemblies within the overall wall assembly). Bond beams, lintels, joints, corbels are examples of well-defined masonry wall features. Other examples might be arbitrary. Because every masonry feature implies specific types of masonry parts, construction processes and functionality, masonry features are considered to be the fundamental semantic units composing masonry wall models. Figure 20 shows the relationship between these categories and main IFC types.

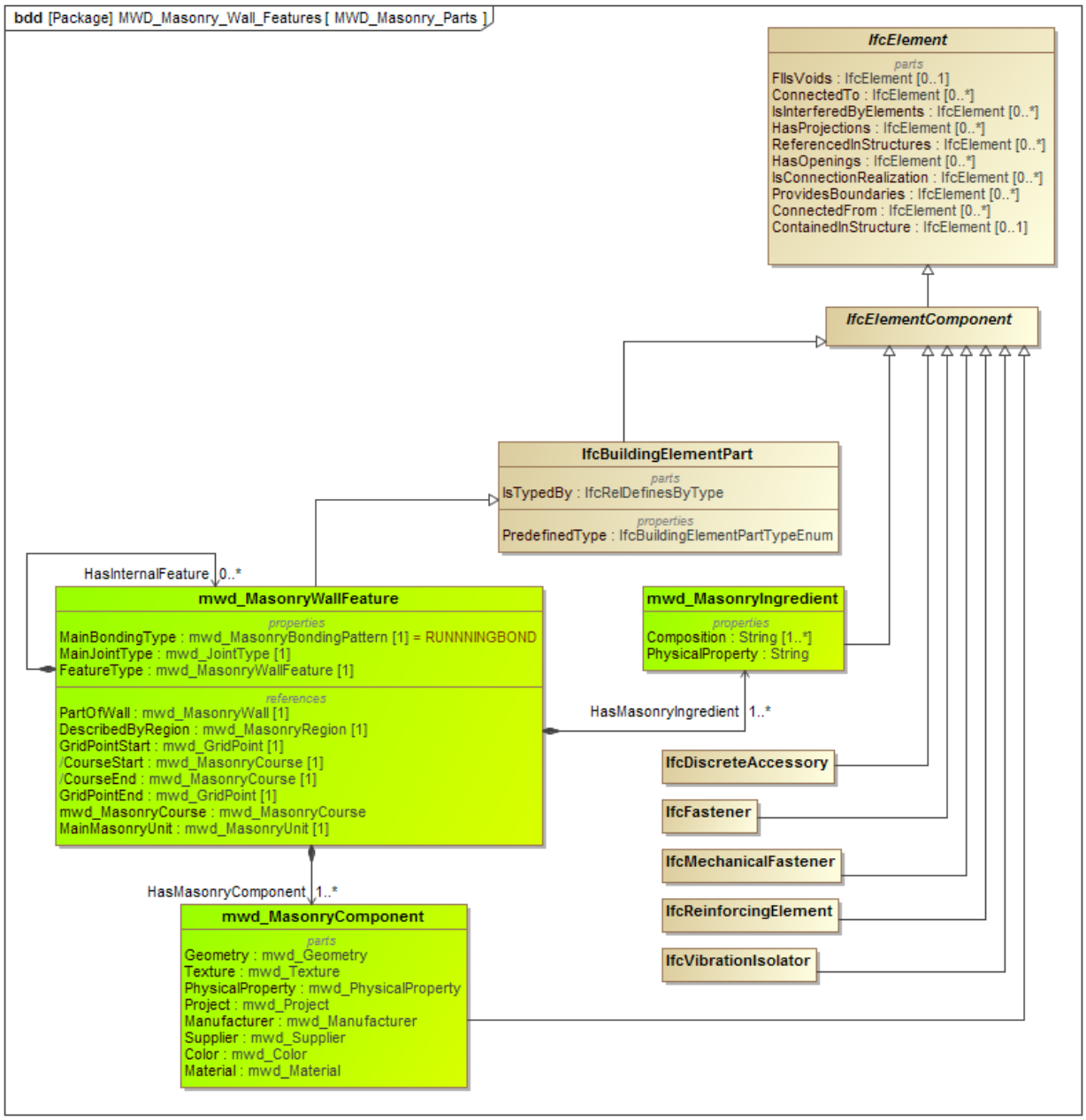

Figure 20 Proposed inheritance structure for masonry features, components and ingredients.

Masonry walls are the highest level of aggregation (i.e. assemblies) of different masonry features (i.e. sub-assemblies), which in turn aggregate smaller masonry components and ingredients (i.e. parts). These aggregation relations across multiple levels of the wall assembly need to be controlled geometrically and dimensionally by a smart modular grid. In terms of geometric representation, masonry features are described by masonry regions with variable levels of geometric detail. The concepts of smart masonry grid, masonry region and the entire masonry wall assembly are defined by the following categories:

- mwd_MasonrySmartGrid (section 3.3.2.4)
- mwd_MasonryWallRegion (section 3.3.2.5)
- mwd_MasonryWall (section 3.3.2.6)

3.3.2.1 Masonry Wall Components (mwd_MasonryComponent)

This category includes all small, discreet components that are used to build masonry walls and that can be acquired off the shelf. The most similar IFC entity for this case would be IfcDiscreteAccessory. However, it was considered necessary to create an sibling category conceptually under ifcElementComponent, so that masonry specific properties could be assigned in the future (possibly through the mechanism of IfcRelDefinesByType, and IfcPropertySet).

This new proposed category mwd_MasonryComponent incorporates many of the general properties already identified for masonry units by the Masonry Unit Definition project (Figure 21). At this point it was considered that those properties apply, at the general level to all masonry components (Figure 22) that are discreet and can be procured off the shelf from catalogues. This allows the future integration between MUD and MWD. Figure 23 shows the inheritance structure from mwd_MasonryComponent down to more specific types of masonry units described by MUD. It is important to point out that list of masonry components presented here is preliminary and non-exhaustive, requiring revision and extension.

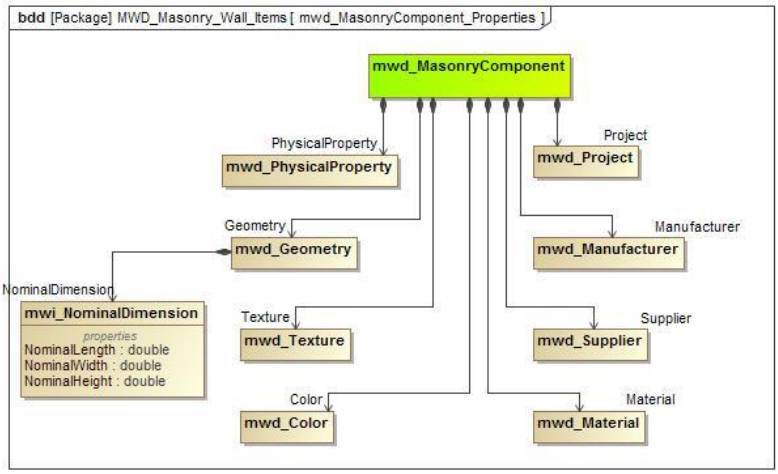

Figure 21 Shared properties for all mwd_MasonryComponent, based on MUD.

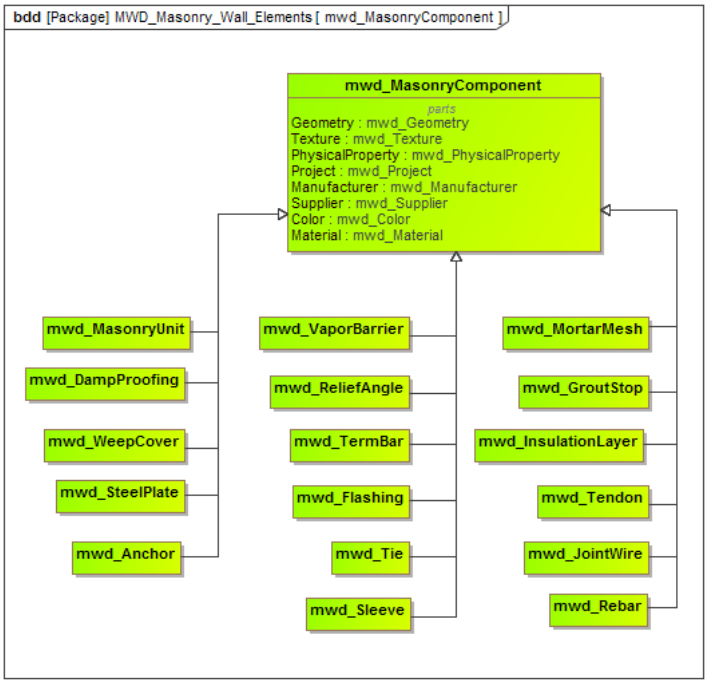

Figure 22 Preliminary list of most common masonry component types.

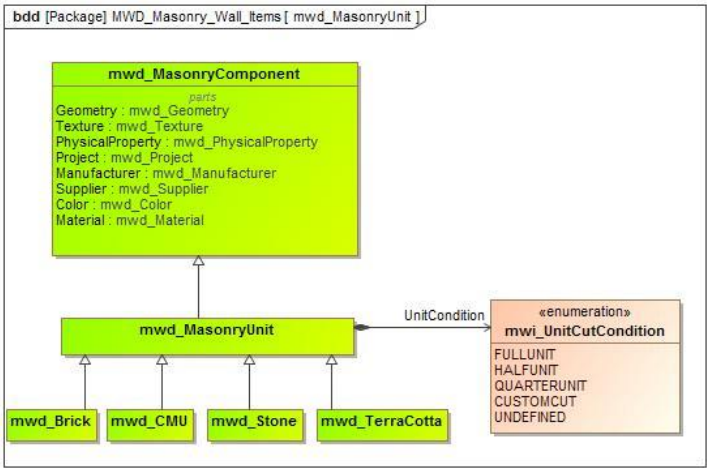

Figure 23 Detail of inheritance structure for mwd_MasonryUnit.
Preliminary list of most common masonry unit component types.

3.3.2.2 Masonry Ingredients (mwd_MasonryIngredient)

This category is proposed for parts of masonry walls that cannot be characterized in the same way as masonry components described in the previous section. That is, all parts that cannot be procured as off-the-shelf discreet units, such as mortar, grout and concrete. On the other hand it was considered that the conventional characterization of these products as 'material' does not provide the proper description of properties that are relevant for masonry construction. More specifically, masonry wall are

defined as assemblies of masonry units plus mortar, and in some cases grout and concrete for reinforcement purposes. Because of that, these parts must also have properties related to assembly sequences, tools, skills and other resources required in similar way than other discreet parts of the assembly. On the other hand requirement associated to on-site preparation makes them very different.

This suggests a different characterization through more specific property sets. For this purpose the name for this category has been suggested as mwd_**MasonryIngredient**, to differentiate it from mwd_**MasonryComponent**, which comprises discreet off-the-shelf parts. In general, an ingredient suggest a chemical combination or mixture (From Merriam Webster: "Full Definition of INGREDIENT: something that enters into a compound or is a component part of any combination or mixture." http://www.merriam-webster.com/dictionary/ingredient).

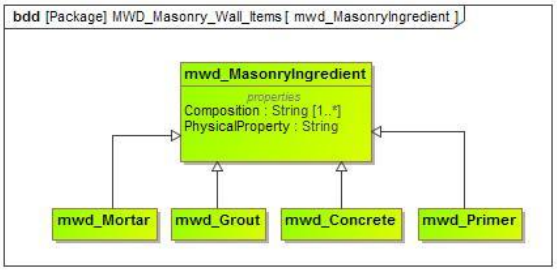

Figure 24 Preliminary list of mwd_MasonryIngredient.

3.3.2.3 Masonry Wall Features (mwd_MasonryWallFeature)

A Masonry wall feature (mwd_MasonryWallFeature) represents sub-assemblies of masonry walls that are result of a particular aggregation of masonry units, components and ingredients. Such sub-assemblies typically have a main function associated and are related to some particular construction process and time sequence. Masonry wall features are defined either explicitly by bona fide boundaries (e.g. a masonry pilaster, bond beams or decorative inlays) or by fiat boundaries. The latter is the case when feature need to be identified arbitrarily within the wall because they have a special purpose for some stakeholder at a specific stage of the wall lifecycle. Examples of this second case may include corner conditions, joints, the courses that a crew of masons can lay by day, or those areas of the wall that are reachable by a given scaffolding method. Notice that in the case of corners and joints, the arbitrary boundaries do not refer to the edge of the corner or the gap of the joint, but to the *extension* of masonry units and components around such edge or gap. For instance, how many bricks of blocks to the left or to the right need to be considered as part of a corner feature?

This type of question is relevant from a representational perspective, and the answer may be variable according to the stakeholder point of view. In some cases the units and components that are within the corner boundaries are just those that interlock at the corner (e.g. stone quoins). In other cases the extension of units to be considered may include all of those that require specific type reinforcement. In other cases the extension of units to be considered are those between the corner edge and any other vertical feature that is in close proximity, indicating a particular assembly process or sequence of erection. Figure 24 illustrates the idea, where the right boundaries of the corner feature is extended until it reaches a control joint.

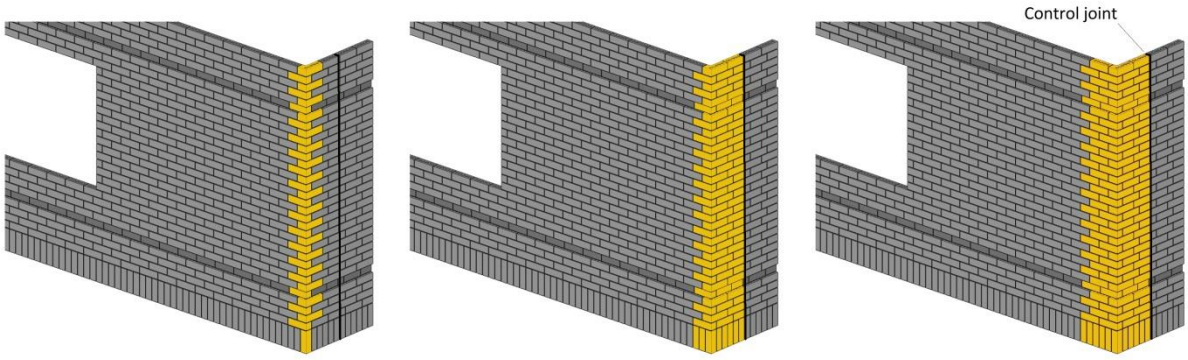

Figure 25 Variable boundaries of a corner.

Because of the criteria variability, there is a need for a mechanism to define customizable boundaries surrounding a masonry feature. This is especially important for those features that do not have self-evident boundaries such as the corner conditions and joints, or areas of the wall that are subject to specific performance requirements (e.g. course laid per day). Additionally, such mechanism needs to represent masonry features in a simplified form so to avoid the cost of representing all masonry units and components associated with it upfront. This idea is similar to a bounding box representation, which works as a proxy or place-holder to be populated with more geometric detail as needed later on the process. Such a place-holder representation proposed here is the masonry wall region (i.e. mwd_MasonryWallRegion), introduced later in section 3.3.2.5.

It is important to reiterate the idea a masonry region is just a proxy representation for the feature's geometry. The real-world entity to be described in the model still is a masonry wall feature, which requires its own semantic definition of properties and relationships. Given the conceptual similarities, an mwd_MasonryWallFeature is proposed as semantically equivalent to a subtype of IfcBuildingElementPart.

The documentation of IFC2x3 describes occurrences of IfcBuildingElementPart as "major components as subordinate parts of building elements". For example, buildingSMART recommends the use of these entities for the representation of layers in sandwich walls, "when the layered material representation does not sufficiently describe the element" (see IfcMaterialLayerSet). From this defintyion it is understood that an IFC part includes sub-assemblies.

In this case, a mwd_MasonryWallFeature is considered a case of IfcBuildingElementPart since it is a major part of the building element of a wall (i.e. IfcWall). The more specific characteristics of a masonry wall feature that differentiates from other wall parts are:

a) All instances of a mwd_MasonryWallFeature represent sub-assembly of masonry units and unit types arranged or combined in a specific way. Such arrangement differentiates the feature from its surrounding masonry bonding pattern. In some cases the masonry feature imply a geometric characteristic that makes the difference from its surroundings visible (e.g. a pilaster, an inlay), but this may not be always the case (e.g. a bond beam).

b) A mwd_MasonryWallFeature may have special constructibility and performance requirements associated. For instance, intricate decorative patterns, complex bonding patterns or reinforcement imply different skills levels, time and resources than other simpler features. A classification of masonry features may contribute to cost estimation among other design and construction activities.

c) Instances of mwd_MasonryWallFeature are geometrically represented by means of a mwd_MasonryWallRegion. A mwd_MasonryWallRegion is an abstract view of the mwd_MasonryWallFeature. In particular, a mwd_MasonryWallRegion represents an abstract view of the masonry units and accessories that conform the feature, in such a way that the feature can be represented with lower levels of geometric detail and complexity.

As a proposed sub-type of IfcBuildingElementPart, occurrences of mwd_MasonryWallFeature inherits top-level properties and relationships such as aggregation / decomposition as well as connections, projections and voids, among others. As a specialization, mwd_MasonryWallFeature is defined by masonry-specific properties, established either by direct or inverse relationships:

Direct relationships

- **HasMasonryComponent**: Masonry wall features are sub-assemblies of masonry units and components that are discreet and procured off-the-shelf. In IFC the most similar objectified relationship is IfcRelAggregates and the inverse IsDecomposedBy. Therefore, HasMasonryComponent is proposed as equivalent to IfcRelAggregates.

- HasMasonryIngredient: Masonry wall features are also made special mixtures prepared on site (e.g. concrete, mortar and grout). In IFC the most similar objectified relationship is IfcRelAssociatesMaterial. Therefore, HasMasonryIngredient is proposed as equivalent to IfcRelAssociatesMaterial.

- **HasInternalFeature**: Masonry wall features can have internal, lower level features. In IFC the most similar objectified relationship is IfcRelAggregates and the inverse IsDecomposedBy. Therefore, HasInternalFeature is proposed as equivalent to a specialization of IfcRelAggregates, which is restricted to objects of the same type mwd_MasonryWallFeature.

- **AssignedToConstructionProcess:** A masonry wall feature has specific construction processes assigned. In IFC the most similar objectified relationship is IfcRelAssignsToProcess. Therefore, AssginedToConstructionProcess is proposed as equivalent to IfcRelAssignsToProcess.

Direct relationships (enumerated types)

- FeatureType: A masonry wall feature may be specified according to an enumeration of most common feature types (See figure 25 for a preliminary list of masonry wall features).

- MainBondingPatternType: A masonry wall feature can have different bonding patterns. However it is characterized by the most prevalent one (i.e. the one that cover most of the feature's volume). This can be specified by an enumeration of most common masonry bonding pattern types.

- MainMortarJointType: A masonry wall feature may have different mortar joint types. However it is characterized by the most prevalent one. This can be specified by an enumeration of most common mortar joint types.

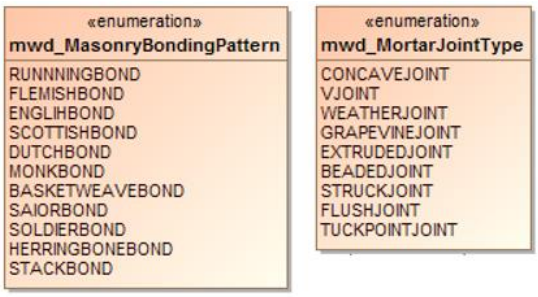

Figure 26 Enumerations for main bonding pattern type and mortar joint type.

Inverse Relationships

- PartOfWall: A masonry feature belongs to a specific wall. In IFC this type of parthood relation is handled by IfcRelAggregates and the inverse IsDecomposedBy. Here PartOfWall is preliminarily proposed as an inverse relationship.

- DescribedByRegion: A masonry feature is geometrically represented by a masonry region.

- GridStartPoint: Insertion point of the feature on the wall, where the point mwd_GridPoint is a point of the underlying grid mwd_MasonrySmartGrid. The grid start point is specified in terms of g[i][j]. It defines the local coordinate system of the feature.

- MainMasonryUnitType: The predominant masonry unit of the feature. While the entire wall may have a predominant masonry unit (e.g. standard brick), specific features may have their own predominant unit (e.g. stone quoins in corners). In IFC this type of parthood relation is handled by IfcRelAggregates and the inverse IsDecomposedBy. Here MainMasonryUnitType is preliminarily proposed as equivalent to a specialization of IsDecomposedBy.

Derived properties

- CourseStart: A derived property that indicated the lowest course (mwd_MasonryCourse) on which the masonry wall feature starts.

- CourseEnd: A derived property that indicated the highest course (mwd_MasonryCourse) on which the masonry wall feature ends.

- SpanningCourses: The number of msonry courses a feature occurrence spans. The minimum is 1 and the maximum is the total number of courses of the wall.

-FeatureArea: The area of the feature projected against the main surface of the wall.

- FeatureVolume: The volume of the feature.

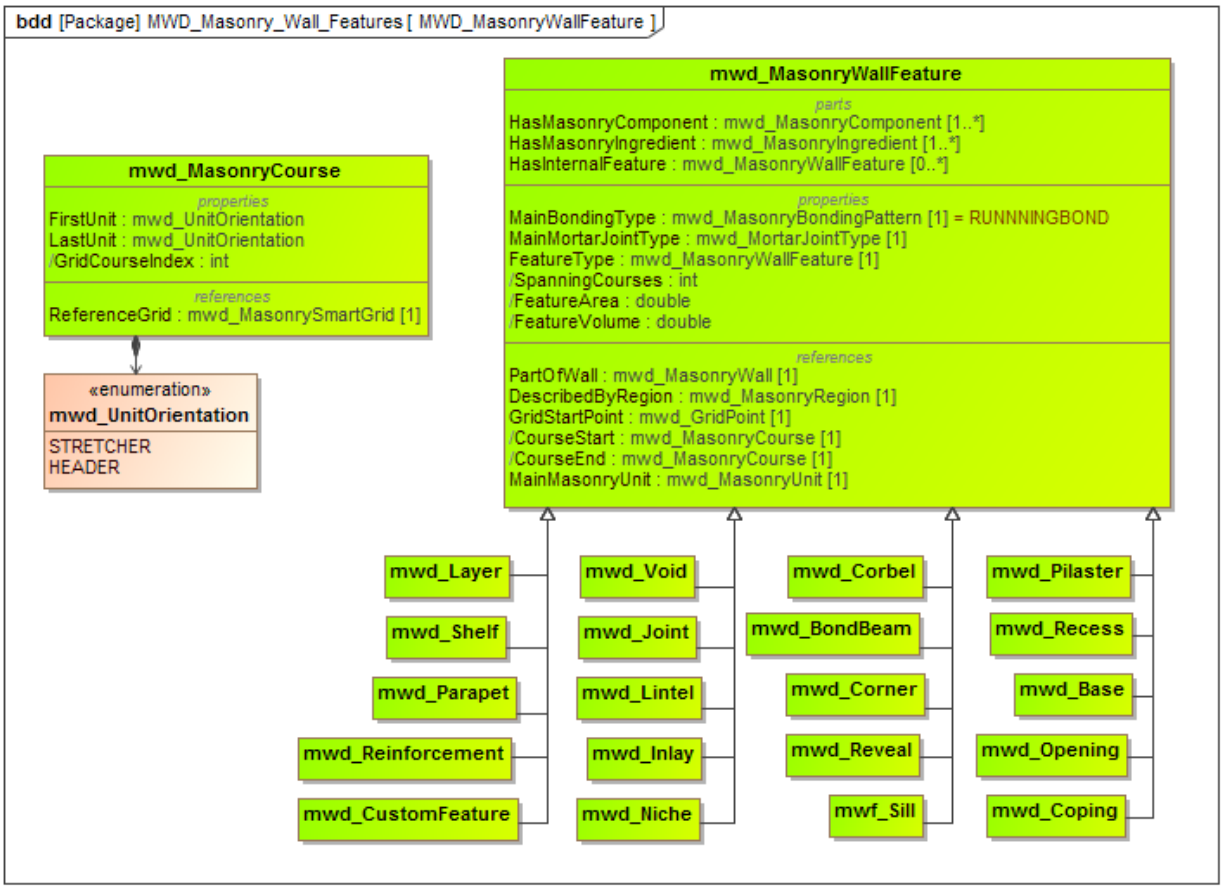

Figure 27 Preliminary list of masonry wall features. Each feature indirectly refers to the masonry wall grid (mwd_MasonrySmartGrid) through its insertion points specified by the properties GridPointStart and GridPointEnd. The courses where the feature starts and ends are derived properties calculated from the insertion points.. Each masonry course knows if its first masonry unit is a header or a stretcher, based on the main bonding pattern of the wall.

The diagram in Figure 26 shows the classes of masonry features that in general define a masonry wall. Masonry features are the fundamental modeling and querying units for a BIM-M model. However, a fully detailed geometric representation of masonry units and accessories composing a masonry feature is time consuming and expensive to model completely. For this reason masonry features will be represented abstractly as masonry regions.

3.3.2.4 Masonry Smart Grid (mwd_MasonrySmartGrid)

A mwd_MasonrySmartGrid is proposed as a virtual grid of smart point objects associated to each individual masonry wall. The main purpose of the grid is to facilitate modular coordination necessary for more effective design and construction of masonry walls. A mwd_MasonrySmartGrid is a quadrilateral grid with a horizontal and vertical spacing between points derived from the nominal dimensions of main masonry unit chosen for the wall. Horizontally, the grid follows the coursing of the wall, defining the centerlines of beds joints. Vertically, the grid defines the centerlines for head joints and the center of stretcher units, since the grid is defined for half the nominal length of the unit. For example, for CMU

units the grid would be 8"x8". Additionally, the mwd_MasonrySmartGrid may be used as reference for the placement of auxiliary local coordinate systems necessary for the positioning of masonry components with an offset from the grid.

The grid is required to enforce dimensional modularity for the entire wall, for the positioning and sizing of internal masonry features and components, as well as for the integration with other systems based on the mechanism of snapping. Another option is to use the grid as a soft reference, which does not enforce modularity a priori, but that can be used later for automatic or manual checking of modular dimensions and coordination.

Points on the grid are intended to be smart objects that are aware of masonry components or building elements that are placed on or close to them. In terms of implementation, a mwd_MasonrySmartGrid may be defined as two-dimensional array of UV points parameterized on a base surface (e.g. IfcPointOnSurface) positioned at one side of the wall. Another approach would be the generation of a collection of curves on that same base surface, spaced according to masonry courses, and the point objects could be propagated along each course line (e.g. IfcPointOnCurve).

In either case any given point could be accessed by means of a two-dimensional index such as g[u][v]. This would give access as well to any given masonry component or building element located on that particular point. However, it is suggested to use a different convention in recognition of masonry wall characteristics. In this regard a masonry specific convention could be g[c][p], where c stands as the index for a masonry 'course' and p stands for 'plumb' position in the horizontal direction of the wall. Other naming conventions could be defined as well. For example, the placement of relief angle for a brick veneer can be specified by an index c, which indicates the particular course on which the relief angle should be attached to. The p index follows the horizontal direction of the wall similarly to the direction that a plumb bob may be moved to verify the wall perpendicularity along different sections. This procedure allows for the identification of individual masonry units and other features within courses. Therefore, for the identification of inner locations in the grid both c and p need to be specified. For example, the end points of a lintel rebar on a CMU wall may be specified by the coordinated g[20][30] and g[20][39] to indicate a rebar that spans 72" (Since the grid is 8"x8").

Given a first masonry unit positioned at g[0][0] as a header or stretcher (lower left corner), and a bonding pattern, it should be possible to determine the location and boundary conditions of all units in the wall. For example, the CMU unit at the bottom left corner of a window opening (looking from outside) located at g[6][8] of a running bond wall is a full block. At g[6][7] is a half-block, since the window was moved half-block to the left. As indicated above, finer granularity (e.g. for 1/4 blocks or custom cuts) can be achieved by auxiliary coordinate systems that are placed locally with an off-set from the master grid point.

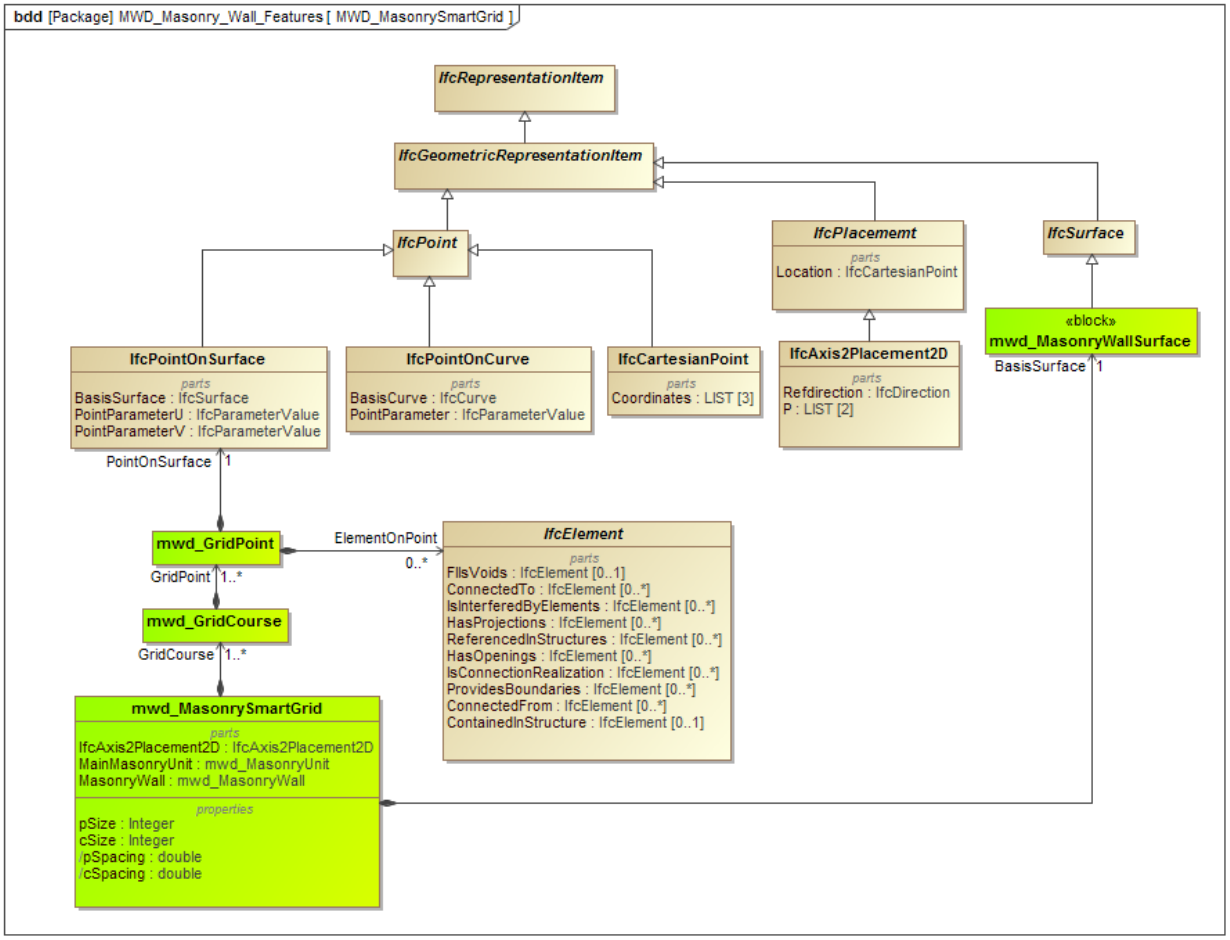

Figure 28 Definition for a smart masonry wall grid.

3.3.2.5 Masonry Wall Regions (mwd_MasonryWallRegion)

A masonry wall region is a representational abstraction of any feature a masonry wall may have. It provides a simplified geometry for the feature, similarly to a solid bounding box. In this way it alleviates the burden of representing the geometry of every single unit and component that is part of the feature, while providing access to the main properties and relationships of the associated feature. These include the list of the masonry units and components that are part of the feature, as well as construction and performance information associate to it (e.g. scheduling, assembly sequencing, load conditions, etc.).

In some cases however, higher levels of geometric information are required. These cases typically involve some form of integration with other building systems, trade coordination or planning for special detailing. For those purposes, a masonry region can also be seen as a place holder for all the masonry units and components that are part of the feature. In this way, additional levels of development can be selectively and incrementally assigned to regions, from LOD 100 to LOD 500 given the information requirements of particular stakeholders and design activities. For that purpose a masonry wall region occurrence has a description of the LOD associated to it (enumerated type mwd_LOD in Figure 33).

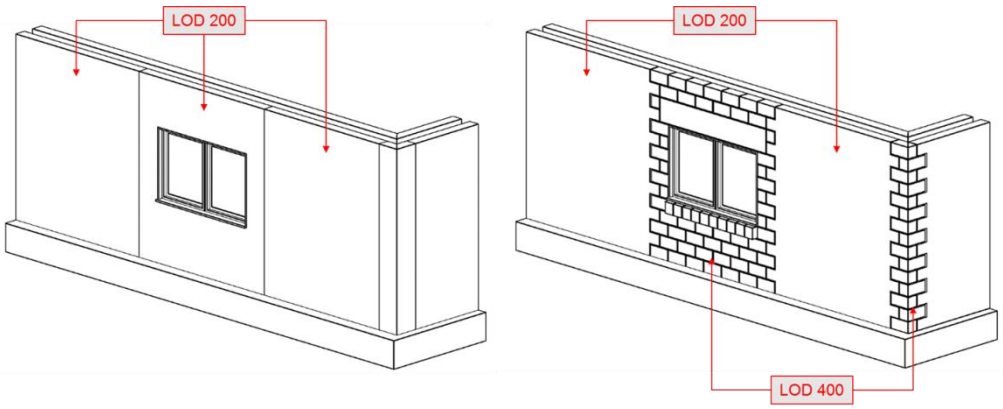

Figure 29 Walls decomposed into opening regions, corner regions and default regions. Additional levels of development can be added selected regions.

Because of its role as geometric placeholder, it was considered that the concept of masonry wall region is semantically equivalent to a subtype of IfcBuildingElementProxy explained in the section 3.3.1.8. This has the advantage of allowing the definition of masonry specific properties and relationships, while keeping the semantics associated to an element proxy, especially in terms of placement and shape representation, among other inherited properties.

3.3.2.6 Masonry Walls (mwd_MasonryWall)

The proposed masonry wall type is a specialization of IfcWallElementedCase, which stands to masonry specific relationships and properties introduced in this report. The inheritance structure proposed for masonry walls (mwd_MasonryWall) and its relationship to masonry wall grids (mwd_MasonryWallGrid), masonry wall features (mwd_MasonryWallFeature) and masonry wall regions (mwd_MasonryWallRegion) is shown in Figure 29. This last figure completes the overall description of the proposed masonry wall schema. This schema is an attempt to provide an comprehensive definition abstractions necessary for the representation of masonry walls within an unified conceptual framework. As such it is still preliminary, and further research and development will be needed by partners of the BIM-M initiative.

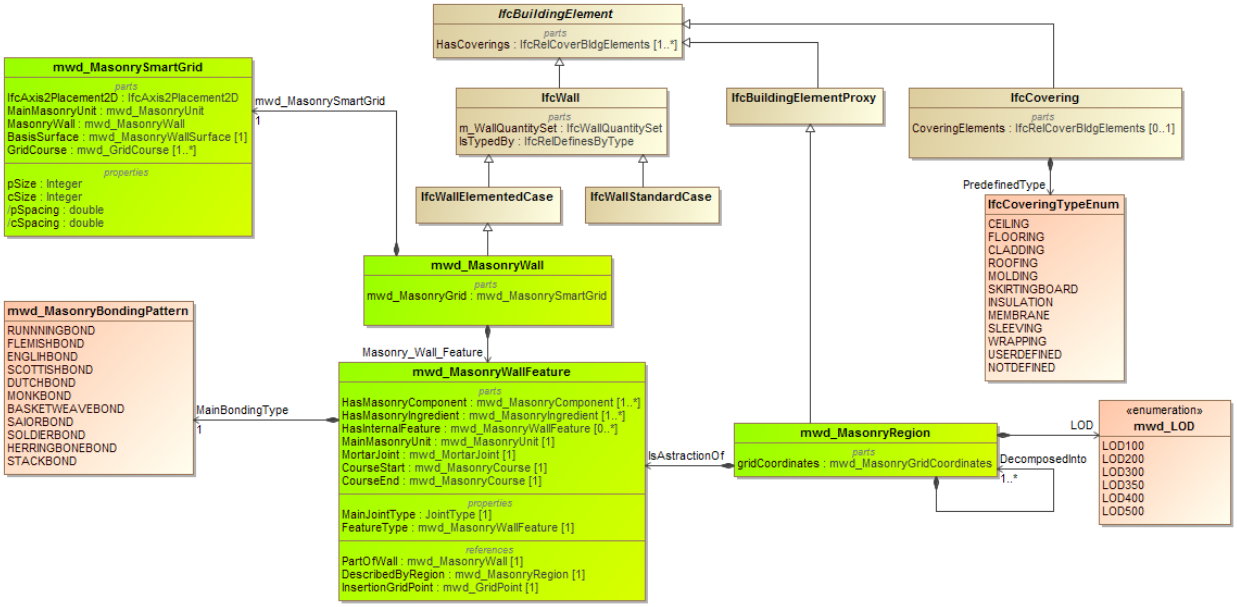

Figure 30 Relationships between mwd_MasonryWall, mwd_MasonryWallGrid, mwd_MasonryWallFeature and mwd_MasonryWallRegion. A mwd_masonryWallRegion is an abstract, proxy representation of a masonry feature that has a LOD associated to it.

The next section provides an overview for the basic functionality expected from BIM application

3.4 Specification of Software Basic Functionality

3.4.1 Software Architecture

The software functionality proposed here is to be implemented as a plug-in for a BIM platform that provides general design services. These include platforms such as Bentley Architecture, Revit, Archicad or Sketchup. This plug-in is envisioned primarily as supporting the architectural design process, working as a central hub for coordination with other disciplines involved. Later on, discipline specific applications can be developed for structural analysis, cost estimation and construction planning of masonry structures. These discipline specific applications can be developed either as plug-in within the same BIM platform or as standalone software.

Since there are a few standalone applications already available in the market, such as Tradesmen Software and CadBLOX for cost estimation, as well as a number of structural analysis applications with specialized masonry modules (e.g. Bentley RAM, RISA), the specification of software functionality proposed here needs to be reviewed for compatibility with the representational requirements of these applications. At the schema level, it is expected that at some point in the future all major applications dedicated to masonry should comply with a standard data model for the representation and exchange masonry specific information.

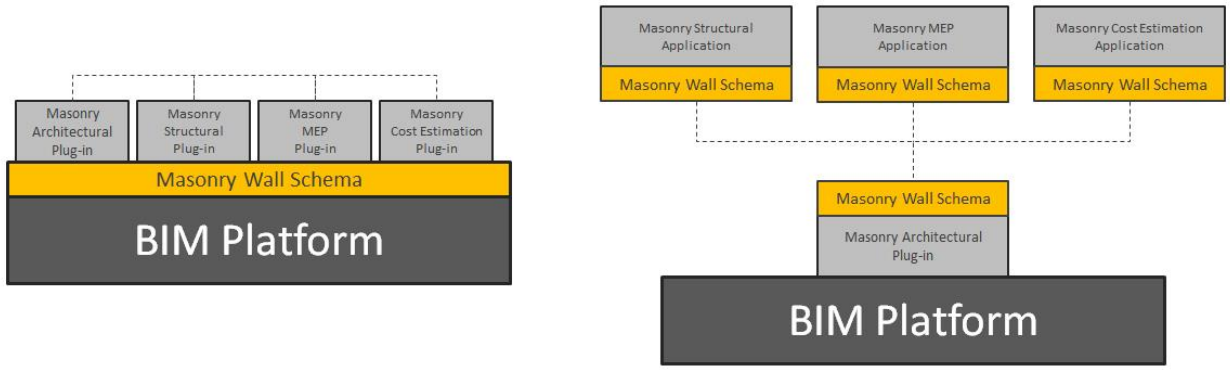

Figure 31 Plug-in architecture for internal or external application integration. On the left a plug-in architecture. On the right, independent application are integrated through a common schema.

3.4.2 Overview of general functionality

The main focus of the proposed software functionality has been on two main aspects:

1. The definition of a set of basic geometric operators to allow the creation and management of masonry wall models at various levels of detail. The description of these geometric operators has been made at a very generic level, so that it could apply to different domain specific applications (e.g. architectural design, structural analysis, cost estimation, construction planning, etc.). More specific functionality for each domain needs to be developed based on individual requirements, ideally as extensions of the basic functionality proposed here.
2. The different geometric objects created by the software application needs to conform to the semantics of the masonry specific schema introduced in sections 3.2 and 3.3. In fact, much of the geometric operators are already available in many BIM applications, but without masonry specific semantics associated. An example of this is the recent functionality added in Revit 2015 of transforming single solid walls with embedded layers into assemblies of discreet parts. It is the intent of this research to provide the criteria for these discretized parts to acquire proper masonry semantics according to the Levels of Development required at different design phases.

3.4.3 Functional capabilities

The functional capabilities envisioned for a BIM software to support the creation and management of masonry wall information are based on the concept of masonry wall features. As explained in section 3.3.2.3, a masonry wall feature represents the fundamental semantic unit behind masonry wall models. For instance, each masonry wall feature is positioned at a specific location within an overall masonry grid, has an overall shape and dimensions and is made of a unique set of unit types and components. Such information may be associated to specific types of construction processes and resources (e.g. equipment, skilled labor, etc.), which in turn have implications on other construction activities, timelines and cost.

The next subsections explains the proposed functionality following a top-down modeling approach for the creation of a generic masonry wall. The process start with the definition of a modular grid, followed by the incremental addition of detail by decomposing a masonry wall into smaller regions, each representing a masonry feature.

For that purpose a simple sandbox prototype was developed in SketchUp by importing a XSD[3] version of the schema introduced in section 3.3.2 (mwd.xsd). The imported XSD file allowed us to characterize the geometric components of a masonry wall modeled in SketchUp according to the semantics defined for masonry regions and features.

The masonry wall modeled is a cavity wall with brick veneer and CMU backup system. This model was developed based on the masonry detailing series provided by the International Masonry Institute (www.imiweb.org) and available through the SketchUp 3D Warehouse. Figure 32 shows the initial exercise of modeling the wall at four different LODs, from LOD 100 to LOD 500. Eventually it became clear that LOD 350 could carry roughly the same amount of information than the LOD 500 model, but with a fraction of the time and effort required to model it. Therefore the main examples focused on the semantic characterization of the model at LOD 350. The main masonry features included in the models at LOD 350 and 500 are:

- Corner condition
- Window opening
- Lintel
- Sill
- Jambs
- Expansion and control joints
- Sealants
- Bond beam
- Vertical reinforcement with grouted cells
- Niches for the roof joists
- Parapet
- Openings for mechanical systems
- Wall layers (air cavity, insulation, vapor barrier, primer, etc.)
- Flashing
- Term bar
- Base (foundation)
- Reveals for the masonry veneer
- Base and parapet soldier coursing in base and parapet of veneer layer
- Relief angle

[3] A XSD schema is a schema encoded in XML.

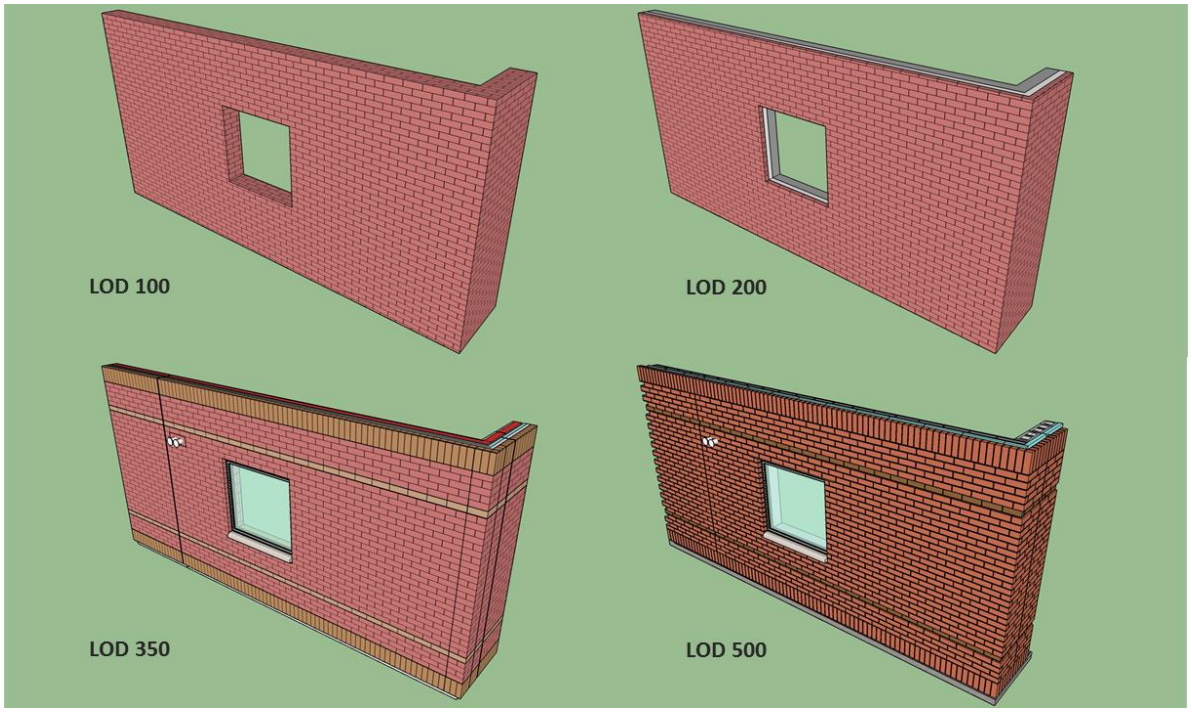

Figure 32 Sequence of LODs for masonry wall model.

Figure33 shows the modeling process at LOD 350. The geometric operators used are the conventional ones used in solid modeling. These include extrusions and sweeps with different combinations of Boolean operators (e.g. union, subtraction and intersection). While the top-down modeling process developed in this example was essentially manual, it is envisioned that many sequences can be automated. Examples of this automation can be seen in various plugins available to Revit for the design wooden frame and precast concrete walls.

For instance, IDAT (IDAT 2015) develops software that automates the decomposition of monolithic wall models into a series of precast panels, including the definition of connections, joints, rebars and lifting points. During the process, relevant attributes and semantic relationships are managed by the software so that complete sets of shops drawings, part lists and scheduling can also be automated. In fact the software supports the entire production workflow, which includes the generation of machine files for the fabrication of the panels and transportation planning (i.e. panel stackup). The process can be seen on https://youtu.be/yL8DRt7knNw.

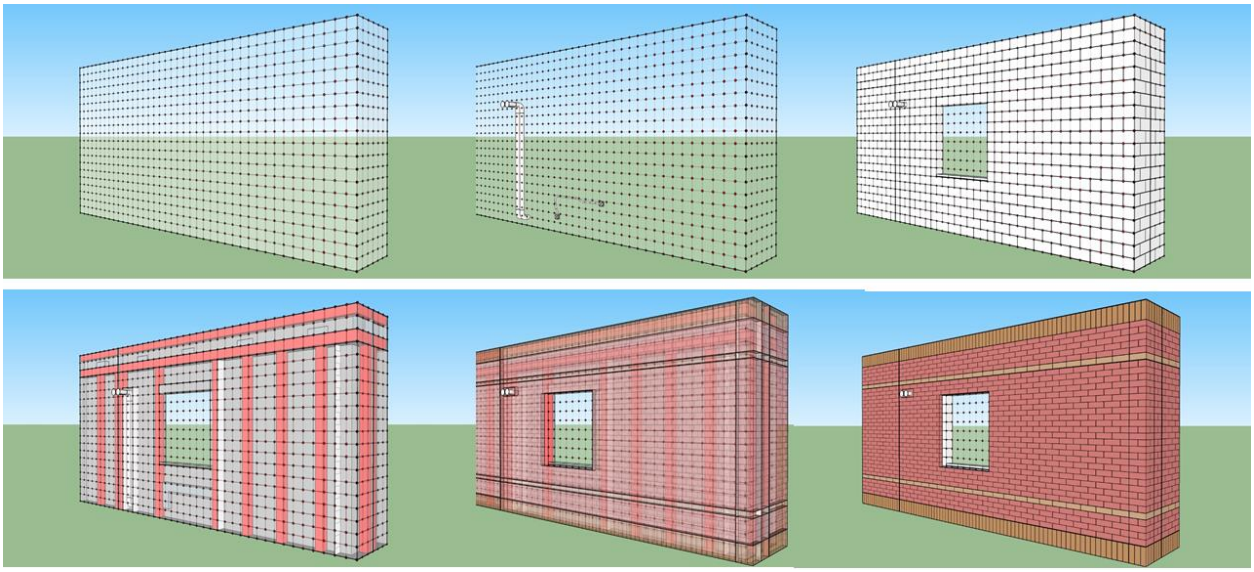

Figure 33 Modeling sequence for LOD 350.

In order to achieve a similar level of automation in masonry, it is important to clarify what are the attributes and semantic relationships among model elements that need to understood and managed by the BIM software. Geometric operators and rules can be implemented accordingly once the semantics of masonry elements have been defined.

The following subsections describe a subset of attributes and sematic relationships identified in this research as potentially relevant. These have been encoded as a XML schema introduced in section 3.3, and preliminarily tested in the SketchUp model showed in Figure 33.

3.4.3.1 Masonry Smart Grid

A masonry wall has to be created with an underlying masonry grid. Such grid can be specified by the user and applied to a wall object before or after its creation. The modular dimensions of the grid are result of the dimensions of main masonry unit chosen for the wall, as provided by MUD (see section 3.3.2.4 for more details). Figure 32 illustrates the basic workflow in setting a Masonry Smart Grid. Figure 33 focuses on the internal activity of selecting the masonry unit to be used from MUD.

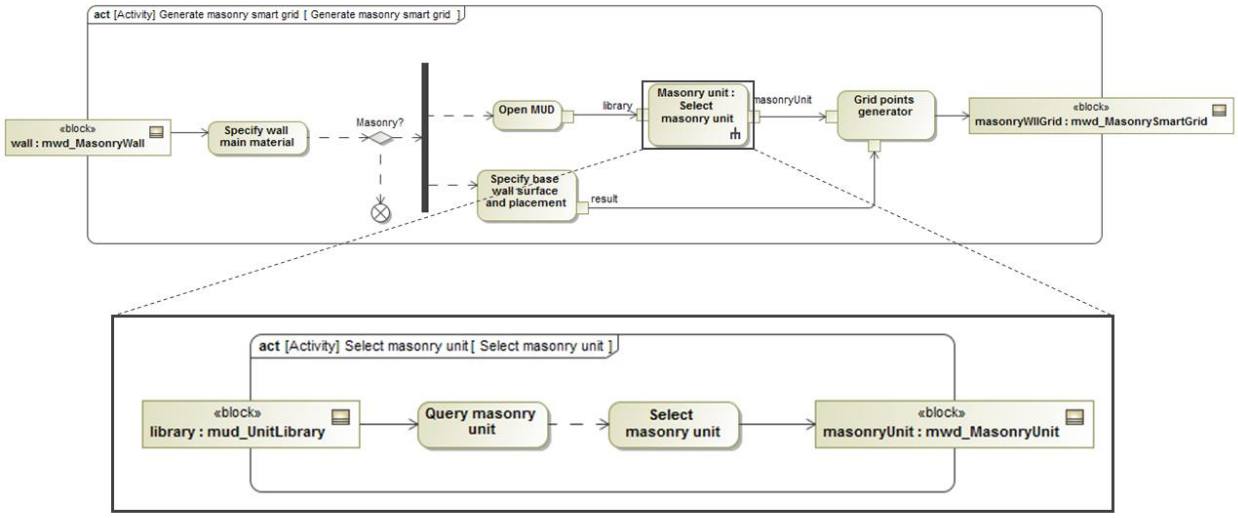

Figure 34 Overall workflow in setting a Masonry Smart Grid. Bottom half of the figure shows the internal activity of selecting a masonry unit from MUD.

Figure 33 shows one possible implementation based on an underlying surface representation of the grid. The surface allows positioning and indexing of smart points according to UV coordinates. As the wall gets stretched, the grid gets extended or reduced accordingly. Smart point objects positioned at the grid determine the placement of masonry wall features.

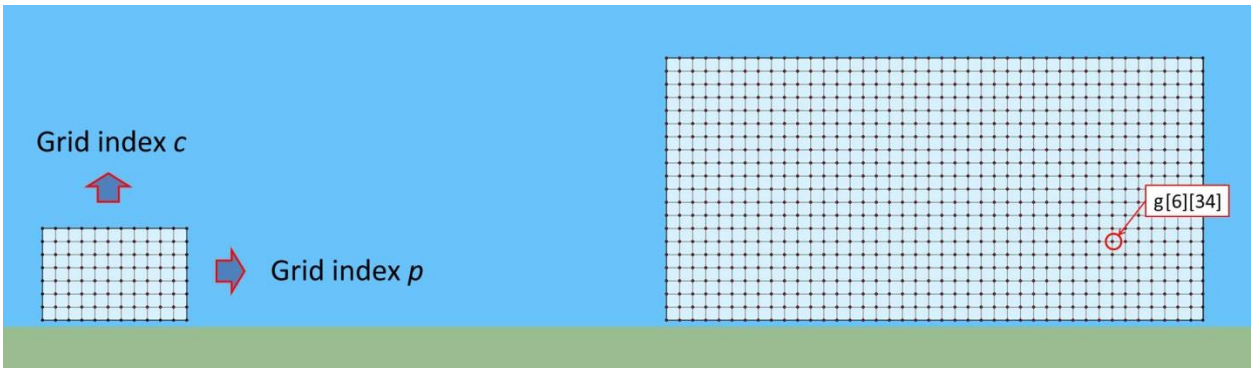

Figure 35 Definition of a Masonry Smart Grid object. The underlying geometric representation could be a surface with a UV mesh for indexing the position of smart point objects.

3.4.3.2 Openings

The underlying masonry smart grid (mwd_MasonrySmartGrid) controls the placement of all types of masonry features defined in the mwd_xsd schema. Such control can be imposed beforehand or after the insertion of the feature. The intent is to ensure modular coordination with the dimension of the main masonry unit chosen for the construction of the wall.

In the example below a window opening was inserted in a CMU backup wall, which generates not only the opening itself but also a surrounding region that contains the opening. The region

attributes cStart, cEnd, pStart and pEnd establish the position of the opening region in terms of the points of the underlying controlling grid. Thus the opening region starts at the course 0, plumb 17 (g[0][17]), and it ends at course 17, plumb 27 (g[17][27]). Additionally, the LOD for that region can be defined independently of other regions of the wall. In this case it was set to 350 (Figure 36).

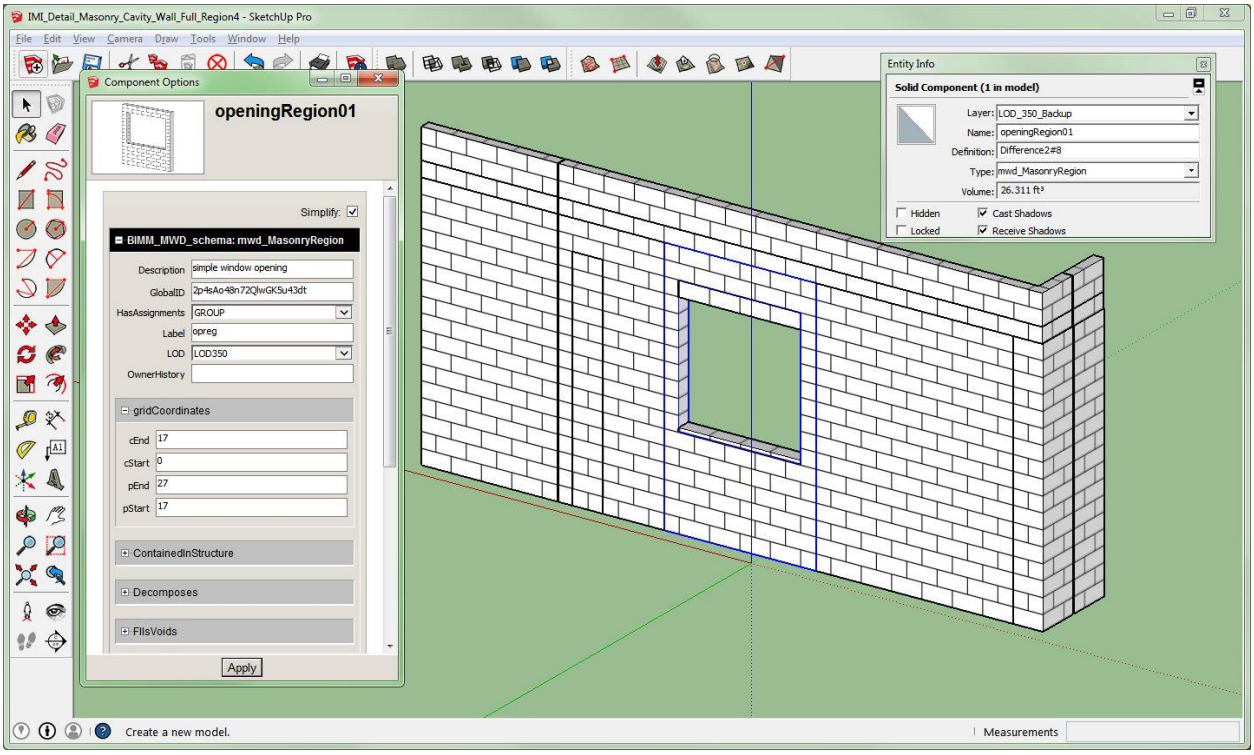

Figure 36 Characterization of a window opening in a CMU backup wall according to the Masonry Wall Definition schema.

The MWD schema allows for further semantic characterization of the opening region. In this case any type of IFC assignment relationship, such as groups, actors, processes and resources can be assigned to the region, as would be expected from any IfcBuildingElement instance. Additionally, the region can be specified according to the masonry feature type it represents, the main masonry unit selected and its main bonding and mortar joint types. Finally, the masonry region can specify if starts with a header or a stretcher unit in its main insertion point.

Figure 37 shows the main attributes and relationships defined for an opening feature in the MWD schema. The region gets formally associated to the opening feature by means of the "*isAbstractionOf*" relationship. The main bonding pattern gets established as well as the main mortar joint type. Additionally, there must be a specification of the first, starting course and the last, ending course for that region. In this case only the start course is showed, which indicates that region starts with a header block (or a half-block) depending if the region is continuous to the adjacent one on the left. Notice that other fields left blank are inherited from IFC and might not be relevant at this stage.

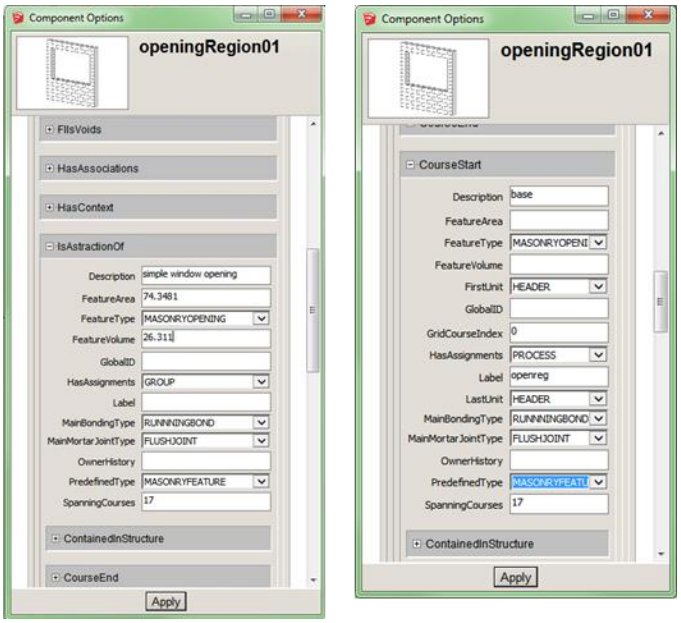

Figure 37 Semantic characterization of region.

3.4.3.3 Lintels and sills

Masonry feature can be decomposed into smaller masonry features. In terms of geometric representation this aggregation relation gets depicted by regions within regions. Because a LOD is a property of the region, internal regions can be described at a higher LOD than its container region, or vice-versa.

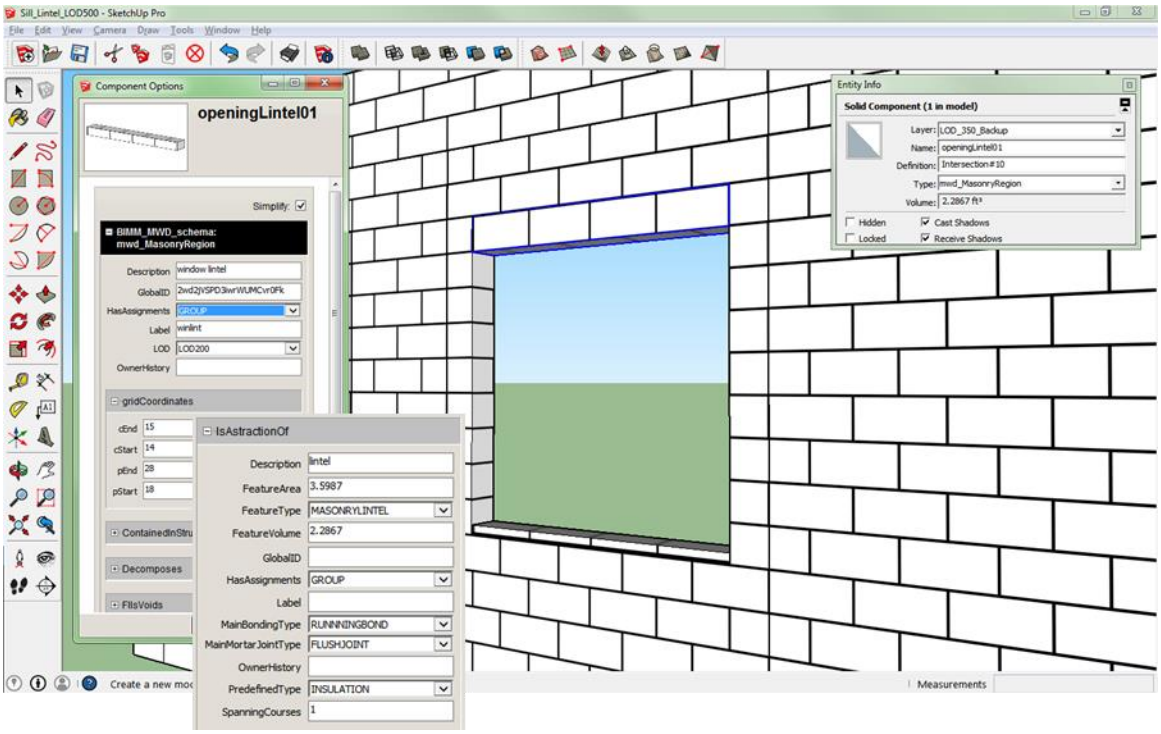

Figure 38 Lintel and sill, depicted by regions at LOD 200.

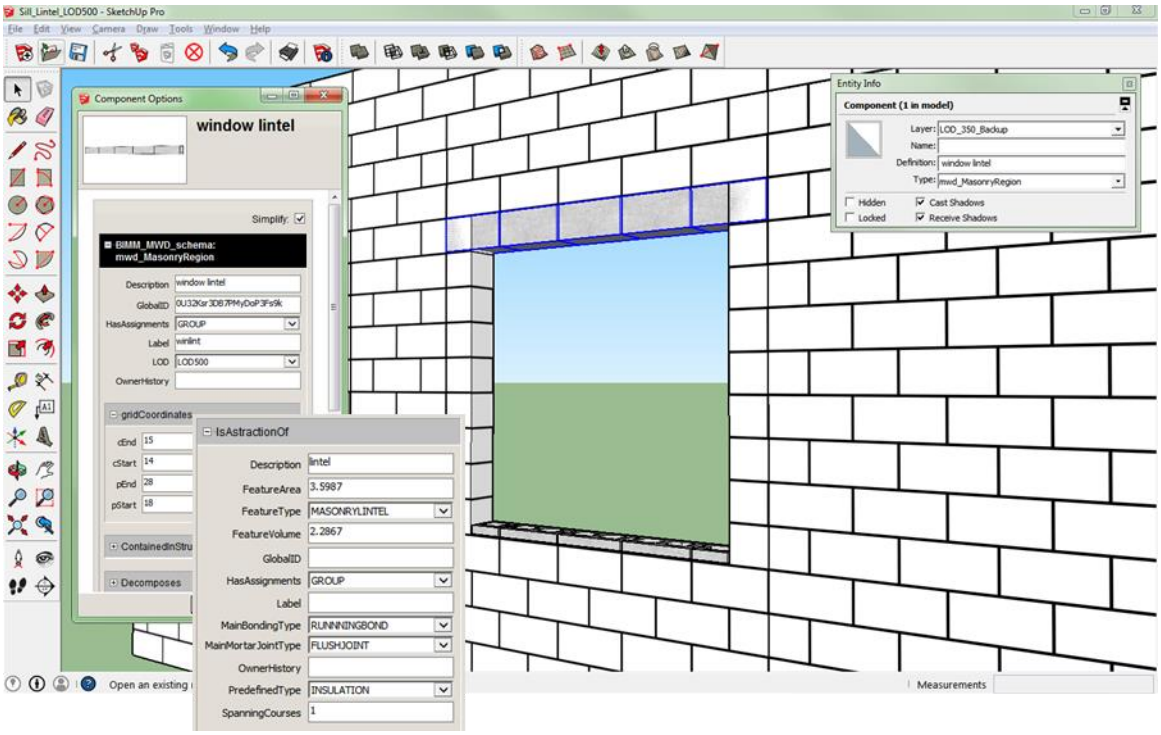

Figure 39 Lintel and sill, depicted by regions at LOD 500. A second set of fields allows the specification of the region as abstraction of a masonry sill (FeatureType property). The uppercase MASONRYFILL is part of an enumeration of predefined masonry feature types.

3.4.3.4 Corners

The representation of corner conditions in masonry walls have been discussed in section 3.3.2.3. Here, only the most common situation is described for a CMU backup wall, using a running bond pattern. The horizontal extension of the corner itself was restricted to one stretcher block on each direction, while the height was restricted to the seventeenth course, since a bond beam was place above that course. Figure 40 shows the corner condition.

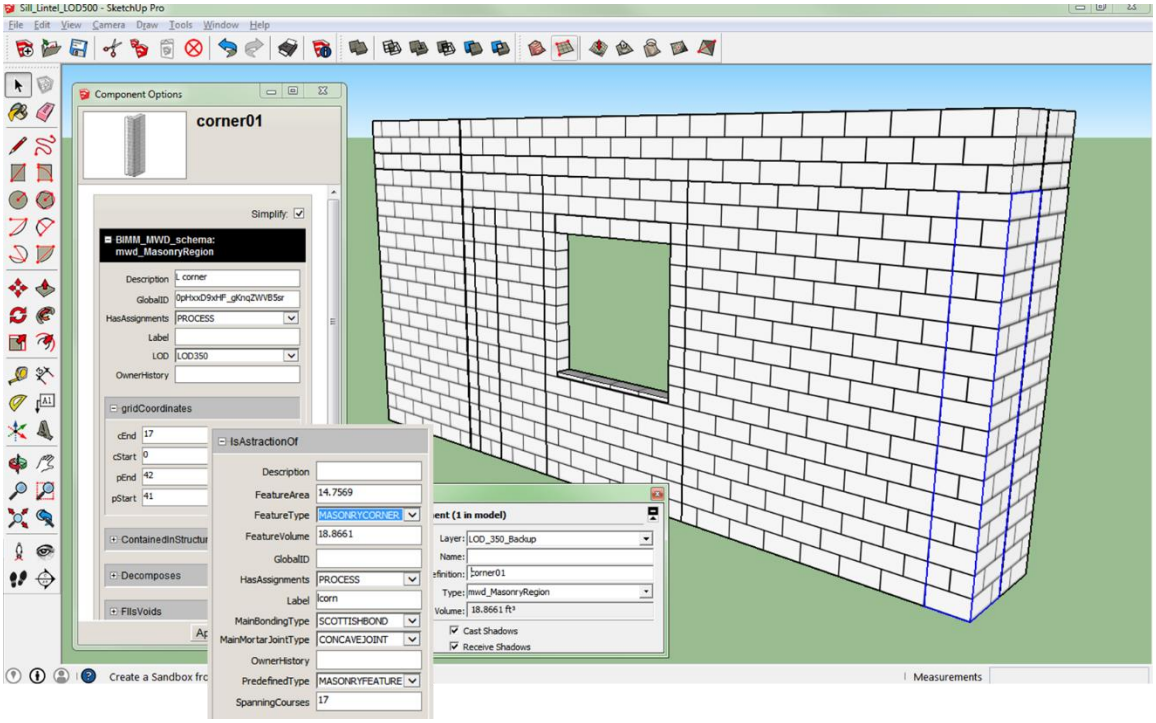

Figure 40 Region representation for a corner feature.

3.4.3.5 Niches

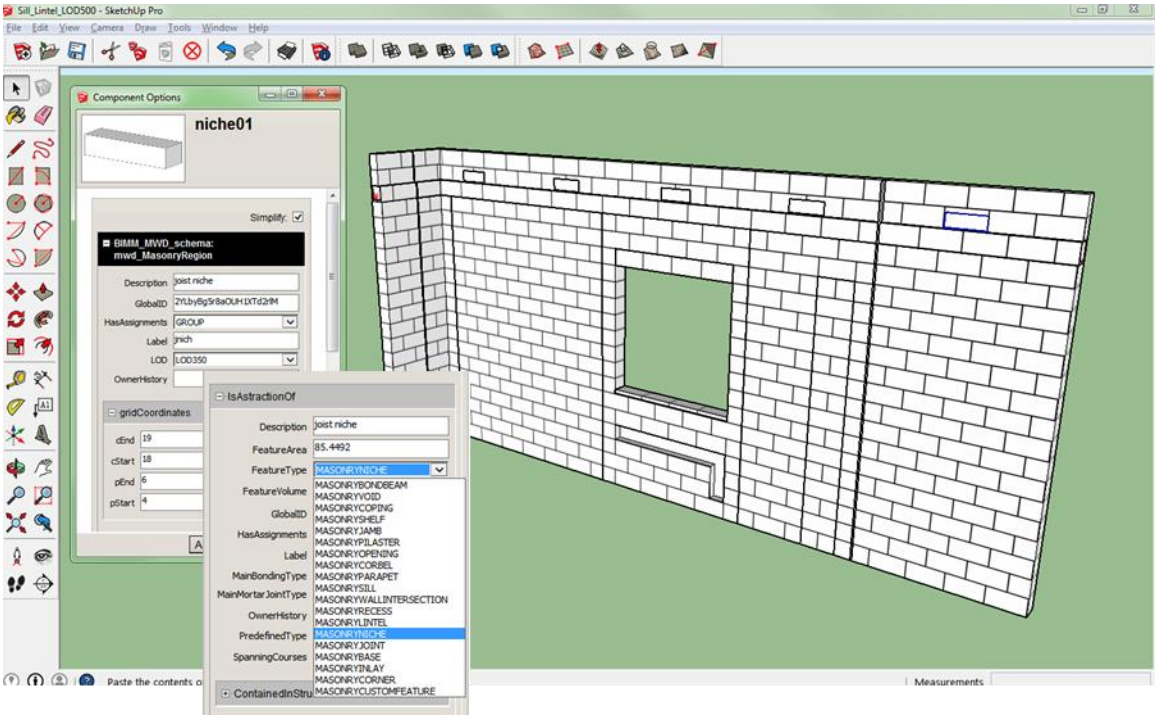

Figure 41Region representation for niche to receive roof joists.

3.4.3.6 Expansion and control joints

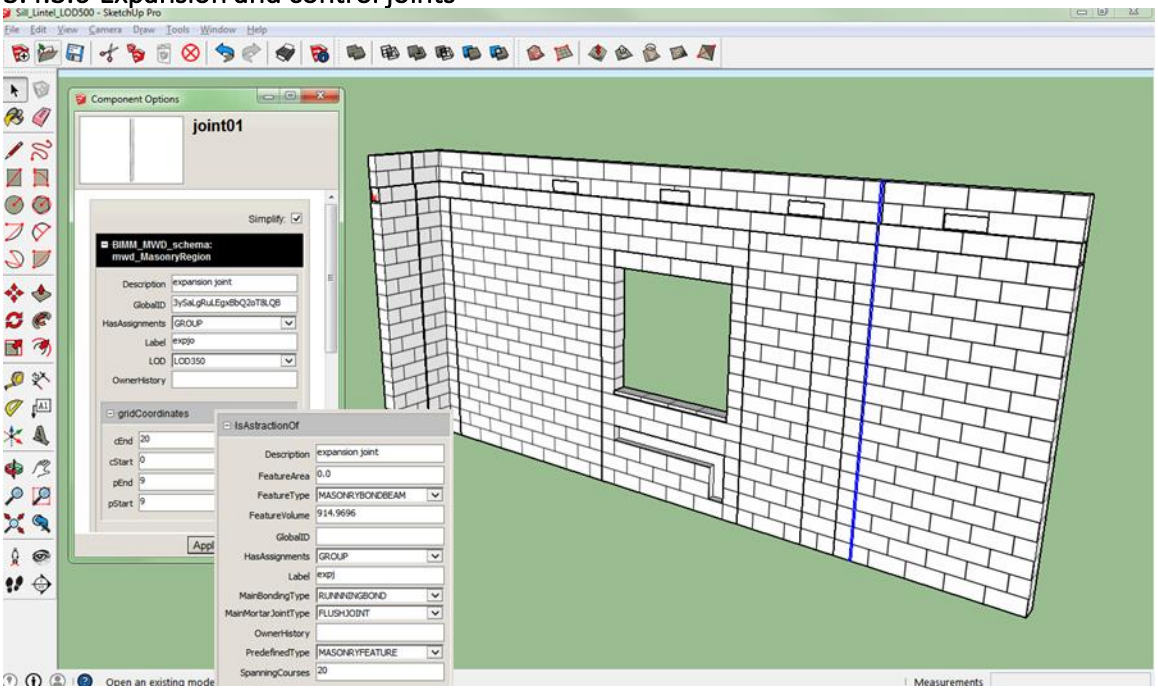

Figure 42 Region representation of an expansion joint.

3.4.3.7 Reinforcement

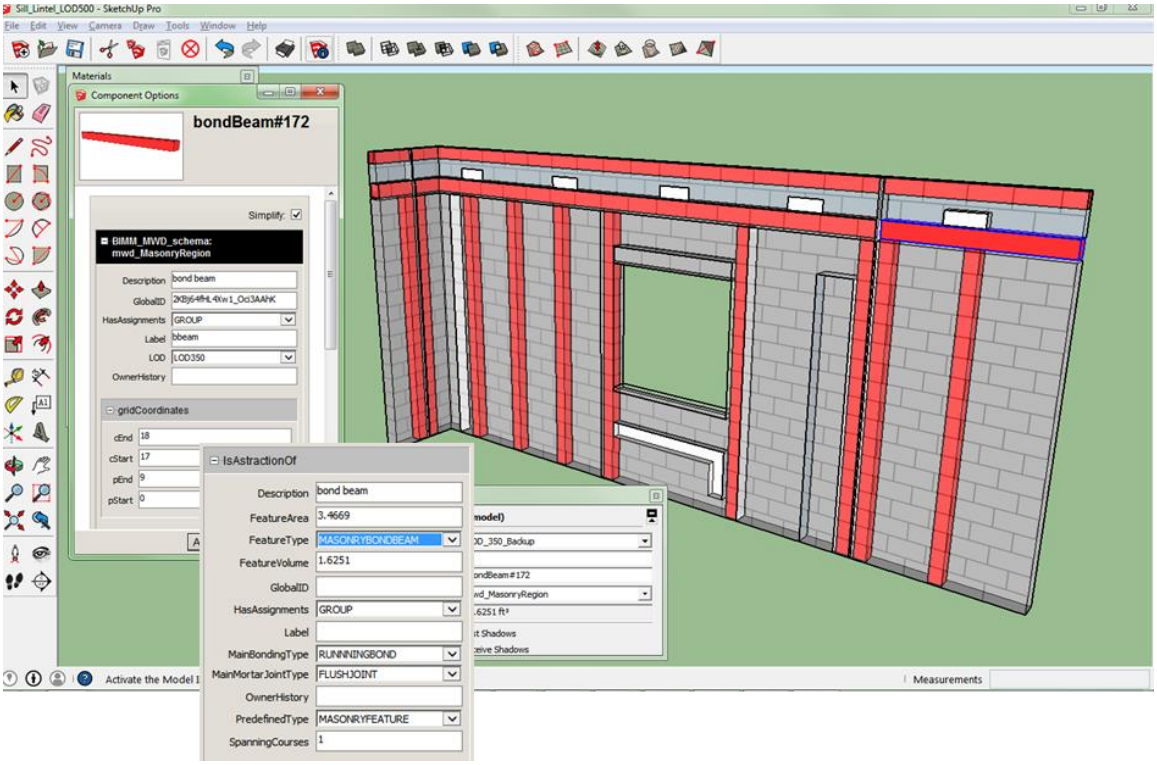

Figure 43 Region representation of a bond beam section.

3.4.3.8 MEP

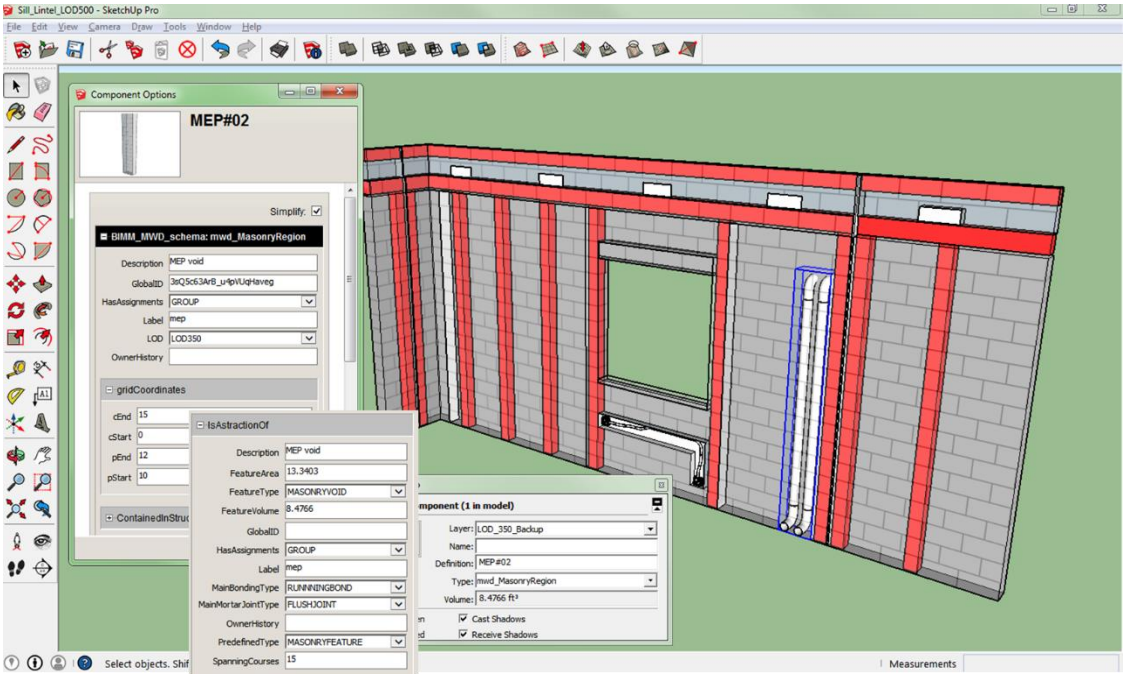

Figure 44 Region representation of a MEP void.

4. SUMMARY

This document provides our underlying philosophy for a compact but extensible representation of masonry walls in BIM platforms. We believe that the schema as described here can be overlain on existing BIM / 3-D modeling platforms (e.g. Revit, Sketch-Up) as well as work with the proprietary wall representations used in masonry specific software packages such as Tradesmen's and CAD BLOX.

The document provides examples of masonry wall representations in IFC, which is the most widely-used non-proprietary format for expressing and sharing BIM models. We expect that this document will be the basis for additional schema development in XML and for the development of software-specific "plug-in" specifications that will be delivered during Phase III of the BIM-M project.

This report is the result of a short six month research activity to formalize and document the thinking of the Georgia Tech team, which has been evolving over the last three years during Phases I and II of BIM-M. We will be requesting detailed reviews of this document from our software collaborators and will be updating the document based on their feedback.

5. REFERENCES

Witthuhn T., Gentry T. R., Sharif S. and Elder J. (2014). Masonry product models for Building Information Modeling, 9th International Masonry Conference, 7-9 July 2014, Guimaraes, Portugal.
Florez L. Castro-Lacouture D. and Gentry R. (2014). Workflows in masonry construction: Analysis of Labor Requirements, 9th International Masonry Conference, 7-9 July 2014, Guimaraes, Portugal.

Gentry R., Eastman C., Haymaker J., and Lee B. (2014). Development of systems models of stakeholders and exchanges involved in masonry construction projects, Report , Project 2, Masonry BIM Benchmark, Building Information Modeling for Masonry, Phase II Project Pankow Foundation.

American Institute of Architects. (2013). Building information modeling protocol form, In AIA Contract Document G202-2013: American Institute of Architects.

BIM_Forum. (2013). Levels of development specification for BIM models, BIM Forum.

Cavieres A., Gentry R., and Al-Haddad T. (2011). Knowledge-based parametric tools for concrete masonry walls: conceptual design and preliminary structural analysis, Automation in Construction, 20, no. 6 : 716-728.

Eastman, C., Lee J., Jeong Y., and Lee J. (2009). Automatic rule-based checking of building designs, Automation in Construction, 18, no. 8 : 1011-33.

Elmasri, R., and Navathe S. (2006). Database Systems. 5th ed. Boston: Pearson Addison Wesley.

Gentry R., Eastman C., and Biggs D. (2013). Developing a roadmap for BIM in masonry: A national initiative in the United States." In The Twelfth Canadian Masonry Symposium. Vancouver, Canada.

Hietanen, J. (2006). IFC Model view definition format, edited by International Alliance for Interoperability.

Ram AdvanSE Version 9.0. (2007). User manual, Bentley Systems Inc. Structural Analysis and Design Software.

Kim, K., and Teizer J. (2014). Automatic design and planning of scaffolding systems using building information modeling, Advanced Engineering Informatics, 28, no. 1: 66-80.

Lee G., Sacks R. and Eastman C. (2005). Specifying parametric Building Object Behavior (BOB) for a building information modelingsSystem, Automation in Construction, no. 15: 758 - 76.

Lee, G., Sacks R. and Eastman C. (2007). Product data modeling using GTPPM – A case study, Automation in Construction 16, no. 3: 392-407.

Lee J. (2011). Building Environment Rule and Analysis (BERA) language and its application for evaluating building circulation and spatial program, Ph.D. Dissertation, Georgia Institute of Technology.

Smith, B., and Mark D. M. (2003). Do mountains exist? Towards an ontology of landforms, Environment and Planning B: Planning and Design 30, no. 3: 411-27.

Smith, B., and Varzi A.C. (2000). Fiat and bona fide boundaries, Philosophy and Phenomenological Research 60, no. 2: 401-20.

Venugopal M, Eastman C., Sacks R, and Teizer (2012). Semantics of model views for information exchanges using the Industry Foundation Class schema, Advanced Engineering Informatics 26, no. 2: 411-28.

Zhang S, Teizer J., Lee J., Eastman C. and Venugopal M. (2013). Building Information Modeling (BIM) and safety: Automatic safety checking of construction models and schedules, Automation in Construction 29, no. 2: 183-95.