

Modeling Semantics for Building Deconstruction

Alberto Paoluzzi¹, Christian Vadalà¹, Danilo Salvati¹, Federico Spini², Enrico Marino²,
Michele Vicentino³, Antonio Bottaro³

¹*Department of Mathematics and Physics, Roma Tre University, My Street, MyTown, MyCountry*

²*Department of Engineering, Main University, MySecondTown, MyCountry*

³*Geoweb s.p.a., Main University, MySecondTown, MyCountry*

{paoluzzi, vadala, salvati, spini, marino}@ing.uniroma3.it, {mvicentino, abottaro}@geoweb.it

Keywords: Building modeling, BIM, Deconstruction Semantics

Abstract: aaaa aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa
aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa
aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa
aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa
aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.aaaa aaaa aaa aaaaa aaaaa, aaaa aaaa aaa aaaaa aaaaa.

1 INTRODUCTION

Da Bottaro

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In

hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

2 BIM AND DESIGN FOR DECONSTRUCTION

Da Michele

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas

nas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

3 GEOMETRIC MODELING AND PROGRAMMING TECHNIQUES

Da Alberto

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In

hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

4 DECONSTRUCTION APPLICATION FOR SURVEYORS

Una decostruzione efficiente ed improntata al massimo riuso dei materiali, deve essere necessariamente supportata da una corretta metodologia che guidi l'utente verso una pianificazione e stima dei costi del processo. Verrà presentata di seguito la metodologia individuata e un'applicazione software che ne rappresenti l'implementazione. In particolare questo software focalizza la propria attenzione sulla determinazione dei costi di demolizione, smaltimento e trasporto dei materiali.

Nella stesura della metodologia sono state prese in considerazione software analoghi come SMART-Waste (Hurley,). Questo, permette di ricavare le stime delle quantità dei materiali fornendo una descrizione della tipologia di edificio e della zona in cui è stato costruito. Grazie a queste informazioni, vengono automaticamente riempiti dei moduli che forniscono una rappresentazione aggregata dei dati di interesse. Il nostro approccio alla decostruzione, prevede invece una modellazione geometrica e semantica delle sue componenti. Infatti, il panorama edilizio italiano, è eterogeneo ed è quindi necessaria una modellazione di dettaglio per poter ricavare informazioni sufficientemente precise.

Il vantaggio della nostra metodologia consiste anche nella possibilità di perseguire un approccio *iterativo-incrementale*, in cui ad ogni fase di modellazione può seguire una fase di validazione dei costi parziali ottenuti. Il risultato può essere eventualmente raffinato ciclando su tale processo. Di conseguenza, il software è stato pensato in modo da agevolare tale metodologia incoraggiando una modellazione che preveda diverse fasi.

4.1 Fasi della progettazione

Creazione progetto: La prima fase della modellazione consiste nella descrizione dell'edificio, in modo da fornire indizi fondamentali per una corretta attribuzione della semantica alle componenti. In particolare sono state identificate come caratteristiche fondamentali l'*età presunta*, lo *stile di costruzione*, lo *storico delle destinazioni d'uso* e la *geolocalizzazione*. L'età presunta e lo stile di costruzione permettono di determinare informazioni sui materiali utilizzati; lo storico delle destinazioni d'uso permette di ricostruire le note di pericolosità degli elementi da smaltire – ad esempio ci possiamo aspettare elementi pericolosi in un'azienda chimica – la geolocalizzazione consente infine di ricavare gli impianti di riciclaggio più vicini al sito.

Modellazione edificio: Durante questa fase l'utente descrive il fabbricato utilizzando alcune tipologie di elementi. Per prima cosa viene definito lo *scheletro dell'edificio* (struttura portante), ovvero l'insieme delle travi e dei pilastri. Su di esso vengono poi costruiti i *muri* (partizioni interne), su cui vengono posizionati gli *infissi* (comunicazioni orizzontali). I *solai* e i *pavimenti* (chiusure orizzontali), invece, sono automaticamente generati a partire dalla topologia dello scheletro e dei muri. Infine, vengono collocati vari elementi come *scale*, *ascensori* (comunicazioni verticali) e *tetti* (chiusure orizzontali) terminando quindi la fase di modellazione dell'edificio.

Attribuzione della semantica: In questa fase agli elementi precedentemente inseriti, si attribuisce della semantica mediante *annotazioni*. Vengono definiti i materiali costituenti, che possono essere uno o più ed in particolare si definisce la densità. Ai fini dello smaltimento, è necessario attribuire uno o più codici CER (*QUI CI VA UN LINK AL PARAGRAFO DOVE SONO SPIEGATI I CER?*) ed il grado di pericolosità. Viene inoltre assegnato un attributo che fa riferimento al cronoprogramma di smaltimento delle componenti del fabbricato.



Figure 1: Interfaccia grafica per l'attribuzione di semantica ad un oggetto della modellazione

Visualizzazione in realtà aumentata: Completate le fasi di modellazione e attribuzione della semantica, si può validare l'intero modello immergendolo all'interno di una *point cloud* precedentemente

ottenuta. (*QUI CI VA UN LINK AL PARAGRAFO DOVE SONO SPIEGATI I CER?*). In questo modo si può verificare l'aderenza del modello alla realtà, ripercorrendo eventualmente i passi precedenti se il risultato non è ancora soddisfacente.



Figure 2: A model inside a point cloud

4.2 Risultati finali

Terminate le fasi del workflow e validata la geometria del modello, l'applicazione fornisce una stima del costo di demolizione. Questo rappresenta l'output desiderato dall'utente in quanto permette di capire se le decisioni prese sono corrette o convenienti dal punto di vista economico ed ambientale. Questo report finale si compone da quattro documenti:

- Una stima dei volumi dei materiali di ogni singolo componente, ricavata a partire dalle geometrie definite in fase di modellazione con opportuni calcoli di integrazione.
- Una stima dei costi di demolizione, smaltimento e recupero. Partendo dai volumi, si determinano le masse grazie alle proprietà dei materiali, mentre conoscendo i CER e quindi la modalità di smaltimento, si stimano i costi del conferimento in discarica.
- Una stima dei costi di trasporto necessari per trasferire i materiali dal sito di demolizione alla discarica. Per fare questo si tiene conto della posizione geografica del modello calcolando i percorsi stradali più convenienti.
- Una stima dei tempi previsti per la demolizione completa del fabbricato, collocati su un diagramma di Gantt. Per fare questo si utilizza l'informazione del cronoprogramma attribuito durante la fase di attribuzione della semantica.

5 DESIGN AND ARCHITECTURE

Workflow and requirements described in the previous section have been received in *Metior*, a prototypal application serving as proof of concept. With the aim of maximize accessibility for surveyors, it is strongly web based and runs in all modern browsers. It is built using React by Facebook and an MVC design pattern with *unidirectional data flow* (red,): it ensures the best code maintainability and debuggability by centralizing access to the application state to a single controller.

5.1 UI & UX

The web application presents itself as a simplified CAD, the UI comprises three main areas of interaction: *toolbar*, *canvas* and *sidebar*, as shown by Figure 3.



Figure 3: *Metior* user interface

From the *toolbar* the user can access functionalities related to: 1) project life cycle (new, save, load); 2) project editing (show-catalog); 3) view/interaction mode switching (2D, 3D); interaction mode changing (selecting, pan, zoom).

The *canvas* is the area in which the user can interact with actual model data. It supports two different view and interaction modes: 1) in *2D-mode* model is displayed as a 2D projection from the top, interaction consist of elements insertion, selection and editing (according to specific plugin interaction prototype, see 5.2.2); 2) in *3D-mode* 3D model can be inspected and navigated respectively via trackball or first-person interaction style, while object picking allows for element selection.

The *sidebar* shows properties of the currently selected element. In the properties panel it's possible to view the description of the element, to add/remove metadata and to modify any property. The latter is the interaction that allows the user to associate semantic to each part of the model.

5.2 Plugin-architecture

The application has been designed to provide a small set of core interaction functionalities and to encapsulate the generation logic for architectural components (from the very basic to the most articulated) into specific plugins.

A plugin is a software component that can be seamlessly integrated into the system in order to extends its capabilities. In *Metior*, a plugin represents an architectural element that extends the Building Information Model design. Technically, a plugin represents a *prototype* (namely a “class” in OOP) of a construction element that can be inserted (“instantiated”) into the *canvas*, thus defining a new element, i.e. a new component of the model.

5.2.1 Plugin definition

A plugin is described by the following eight properties: 1) a unique name; 2) a description; 3) a set of metadata; 4) the *occupation type* (one among *linear*, *area* or *volume*); 5) the *placement type* (*inside* or *over*); 6) a set of specific properties mapping the semantic to associate to the plugin; 7) a construction function that returns the 2D representation of the element (in SVG format, to be used in the *2D-mode*); 8) a construction function that returns the 3D representation of the element (in OBJ format, to be used in the *3D-mode*)

5.2.2 Plugins taxonomy

The plugins can be organized according to the *occupation type* and the *placement type*.

In the *occupation type* three different kind of plugins can be identified: *linear*, *area* or *volume* plugins.

The *linear* ones extend in one dimension (unless a radial thickness) (e.g. hydraulic lines, electrical cables). The *area* plugins extend in two dimensions (unless a linear thickness), (e.g. separation elements). They can be divided into *horizontal area* (e.g. floor and ceil), and *vertical area*, (e.g. walls). The *volume* plugins extend in three dimensions. They can be *fixed volume*, (e.g. a piece of furniture) and *scalable volume*, that can be scaled (proportionally or not), (eg. pillars, staircases).

The *occupation type* determines a different way to instantiate and to insert the plugins into the canvas. In particular, in *2D-mode*, *linear* plugins are inserted drawing lines by mean of a drag&drop interaction; the *area* plugins are inserted drawing the bounding-box of the element by mean of a drag&drop interaction; the *volume* plugins are inserted picking the position of the element by mean of a point&click interaction, and

adjusting their dimensions modifying the bounding-box by drag&drop.

The *placement type* determines if the element can be inserted into the canvas in a specific point occupied or not by other elements.

The *placement type* determines the relationship between a new instance of a plugin and instances of other plugins previously added to the model. The relationship can be of two kind: *inside* or *over*.

Plugins belonging to the *inside* category can be added only inside other element (that can be *linear*, *area* or *volume*); e.g., a “window” is a “volume inside vertical area” element, while an “hydraulic line” is a “linear inside horizontal area” element.

Plugins of the *over* category can be added only over other elements (of any type); e.g., a “pillar” is a “volume over horizontal area” element, while an “electric panel” is a “volume over vertical area” element.

In the design phase, an element that doesn’t meet the placement constraints defined by the *placement type* is notified by the system as a visual warning, showing its bounding-box in semi-transparent blinked red color.

5.2.3 Plugin specific properties

Each plugin has a set of specific properties of the building elements that it represent. Each property is defined by: 1) a name; 2) a type, such as “number”, “text”, “boolean”, or “custom”, 3) a value.

According to its type, each property value can be inserted in different ways. For example, a boolean property value is set through a checkbox, while a textual property is set through a text box. The system is designed to accept custom kinds of property. To a custom property it is required to define the component of the UI that permit the user to insert its value. For example, a “color” property can be introduced defining a UI component composed by three text boxes (one for each RGB components), while a “length” property can be introduced defining a UI component composed by a text box for the value and a drop-down menu for the unit of measure.

The specific properties of an element can be edited in the relative panel in the sidebar, once the element is selected in the canvas.

5.3 Plugin Catalog

It is pivotal to provide surveyors with a rich catalog of plugins, to cover all the basic as well as the most advanced modeling requirements. Table 1 reports examples of plugins requirements arranged according to the introduced taxonomy.

	inside	over / free
linear	pipe	electrical-conduit
ver. area	window, door	wall
hor. area	light-panel	ground, ceil
volume	pillar	staircase

Table 1: Plugins example according to taxonomy 5.2.2

5.4 Server-side models generation

Both 3D and 2D model generation has been designed to be asynchronous: the actual result of the invocation of the generation function is not the model itself but rather a *promise* of the expected result. Such a design is important since the computation for model generation may require a while. In the meantime the user must be able to interact with the interface, which in turn must remain responsive. Relying on this architecture, generation of the models can be easily delegated to a server (as shown in Figure 4), thus relieving the client from the burden of onerous computations. The server exposes a REST-like HTTP based JSON API to the client. The plugin span from the client to the server since the 2D and 3D generation functions (“2Dgf” and “3Dgf” respectively in Figure 4) defined by the plugin are actually executed on the server.

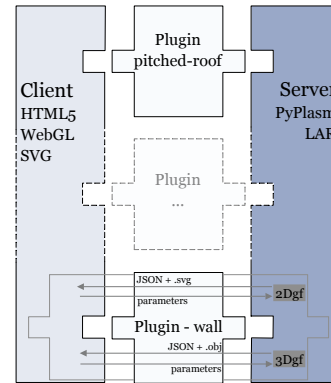


Figure 4: Client/Server architecture for server-side models generation

5.5 A plugin example

POTREBBE ESSERE UTILE INSERIRE LA DEFINIZIONE DEL PLUGIN SCALA AD ES-EMPPIO?

6 CONCLUSIONS

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam

lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

ACKNOWLEDGEMENTS

If any, should be placed before the references section without numbering. To do so please use the following command: `\section*{ACKNOWLEDGEMENTS}`

REFERENCES

- Redux: predictable state container for JavaScript apps.
<http://redux.js.org/>. Accessed: 2016-11-09.
- Hurley, J. W. How to SMARTWaste the Construction Industry. In *10th Symposium Construction Innovation and Global Competitiveness*, Conference Proceedings for the 10th Symposium Construction Innovation and Global Competitiveness.

APPENDIX

If any, the appendix should appear directly after the references without numbering, and not on a new page. To do so please use the following command: `\section*{APPENDIX}`