

# Modeling Semantics for Building Deconstruction

Alberto Paoluzzi<sup>1</sup>, Christian Vadalà<sup>1</sup>, Danilo Salvati<sup>1</sup>, Federico Spini<sup>2</sup>, Enrico Marino<sup>2</sup>,  
Michele Vicentino<sup>3</sup>, Antonio Bottaro<sup>3</sup>

<sup>1</sup>*Department of Mathematics and Physics, Roma Tre University, My Street, MyTown, MyCountry*

<sup>2</sup>*Department of Engineering, Main University, MySecondTown, MyCountry*

<sup>3</sup>*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 (*CITAZIONE\**). 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.



Figure 1: Le fasi della progettazione iterativo-incrementale

## 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.



Figure 2: Rendering 2D della planimetria



Figure 3: Rendering 3D della planimetria

**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.

**Visualizzazione in realtà aumentata:** Completate le



Figure 4: Interfaccia grafica per l'attribuzione di semantica ad un oggetto della modellazione

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.

## 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 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 appears as a simplified CAD.



Figure 5: *Meteor* user interface

L'interfaccia utente si compone di 3 aree principali: - toolbar - canvas - sidebar

Dalla toolbar è possibile richiamare le funzionalità per: - gestire il ciclo di vita di un progetto (nuovo, carica, salva), - editare il progetto (aprire il catalogo di elementi architettonici da disegnare) - cambiare la modalità di visualizzazione (2D & 3D) - cambiare le modalità di interazione (pan & zoom)

La canvas presenta il progetto nelle differenti modalità di visualizzazione - modalità 2D, vista dall'alto, in cui è possibile inserire/selezionare elementi - modalità 3D, vista esterna prospettica o ortografica, in cui è possibile selezionare elementi - modalità 3D, vista in prima persona, in cui è possibile navigare il progetto

La sidebar presenta le proprietà specifiche dell'elemento correntemente selezionato. Attraverso tale pannello è possibile visualizzare la descrizione dell'elemento, modificare le proprietà dell'elemento, aggiungere/modificare/rimuovere metadati.

## 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.

Un plugin è un componente software che può essere integrato nel sistema e che ne estende le capacità.

In Meteor, un plugin rappresenta un elemento architettonico che estende la capacità di progettazione di un edificio. Tecnicamente, un plugin rappresenta un prototipo (in programmazione orientata agli oggetti, una "classe") di un elemento architettonico, che può essere inserito ("istanziato") all'interno di un progetto.

### 5.2.1 Definizione di un plugin

Un plugin viene definito dalle seguenti proprietà:

- un nome univoco
- una descrizione testuale
- metadati
- la tipologia del plugin, che può essere "linear", "area" o "volume"
- le possibilità di posizionamento del plugin, che può essere "inside" o "over"
- le proprietà specifiche dell'oggetto descritto dal plugin
- una funzione che riporta una rappresentazione 2D dell'elemento (per la vista 2D)
- una funzione che riporta una rappresentazione 3D dell'elemento (per la vista 3D)

### 5.2.2 Tassonomia dei plugin

I plugin possono essere organizzati in base alla loro tipologia e possibilità di posizionamento.

Gli elementi di tipo "linear" rappresentano elementi che si estendono in una dimensione (a meno di uno spessore), come ad esempio i condotti idraulici o i cavi elettrici.

Gli elementi di tipo "area" rappresentano elementi che si estendono in due dimensioni (a meno di uno spessore), ovvero gli elementi di separazione. Questi si distinguono in "horizontal", come ad esempio i pavimenti o i solai, e "vertical", come ad esempio le pareti.

Gli elementi di tipo "volume" rappresentano elementi che si estendono nelle tre dimensioni. Questi si distinguono in "fixed", ovvero che hanno dimensioni fisse, come ad esempio i componenti di arredo, e "scalable", ovvero che hanno dimensioni scalabili (in modo proporzionale, o non), come ad esempio i pilastri.

Ciascun tipo determina una modalità di istanziazione ed interazione differente, ovvero una modalità di inserimento e di modifica dell'istanza nella canvas. Nello specifico, nella modalità 2D, gli elementi di tipo "linear" sono inseriti disegnando drag&drop linee, gli elementi di tipo "area" sono inseriti disegnando drag&drop il bounding-box dell'elemento, gli

elementi di tipo "volume" sono inseriti specificando point&click la posizione dell'elemento, e variando le sue dimensioni allargando drag&drop il bounding-box dell'elemento.

Ciascuna possibilità di posizionamento di un plugin determina la fattibilità di inserimento dell'istanza del plugin all'interno della canvas.

Gli elementi di tipo "inside" possono essere inseriti solo all'interno di altri elementi di tipo "linear", "area" o "volume". Ad esempio, una finestra è un elemento di tipo "volume inside vertical area", un condotto idraulico può essere un elemento di tipo "linear inside horizontal area".

Gli elementi di tipo "over" possono essere inseriti solo a fianco di altri elementi di tipo "linear", "area" o "volume". Ad esempio, un pilastro è un elemento di tipo "volume over horizontal area", un pannello elettrico è un elemento di tipo "volume over vertical area".

Nella fase di progettazione, un elemento che non rispetta il vincolo definito dalla sua possibilità di posizionamento è rilevato come warning visivo (bounding-box di colore rosso) dal sistema.

### 5.2.3 Proprietà specifiche dei plugin

Ciascun plugin ha un insieme di proprietà specifiche dell'elemento che rappresenta. Ciascuna proprietà è definita da: - un nome - un tipo, che può essere "number", "text", "boolean", o "custom" - un valore

Ciascun tipo di proprietà descrive e determina una modalità di inserimento del valore specifica. Ad esempio, un tipo "boolean" è definito da una "checkbox", un tipo "text" da una casella di testo. Il sistema permette di definire nuovi tipi di dato, specificando l'interfaccia di inserimento del valore. Ad esempio, un tipo "color" può essere definito da tre caselle numeriche che rappresentano le componenti di colore RGB, oppure un tipo "lunghezza" può essere definito da una casella numerica che rappresenta il valore e da un menu a tendina per specificare l'unità di misura.

Le proprietà di un istanza di un plugin sono editabili dal pannello presente nella sidebar quando l'istanza è selezionata.

## 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 arranged by 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 6), 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 ("3Dgf" and "2Dgf" respectively in Figure 6) defined by the plugin are actually executed on the server.

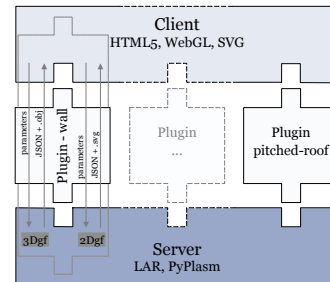


Figure 6: Client/Server architecture for server-side models generation

## 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.

## 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}`