WILEY | Hindawi

Research Article

Procedural Modeling-Based BIM Approach for Railway Design

S. A. Biancardo [ID],[1] **M. Intignano** [ID],[1] **N. Viscione** [ID],[1] **S. Guerra De Oliveira** [ID],[1] **and A. Tibaut** [ID][2]

[1]*Department of Civil, Construction and Environmental Engineering, Federico II University of Naples, Naples 80125, Italy*
[2]*University of Maribor, Faculty of Civil Engineering, Transportation Engineering and Architecture, Maribor 2000, Slovenia*

Correspondence should be addressed to S. A. Biancardo; salvatoreantonio.biancardo@unina.it

Building information modeling (BIM) is a powerful methodological approach for designers that has revolutionized the field of architecture and construction for some years now, minimizing errors and making the entire design, construction, and management process more efficient. The first results have been so encouraging that many countries, from Europe to the United States to Asian countries, have adopted specific regulations to promote its development and use. BIM models are based on the Industry Foundation Classes (IFC) standard, i.e., an object-based file format with a data model developed by building SMART to facilitate interoperability. Objects are characterized by properties, such as geometry, material, cost, and all related construction process information, such as scheduling or the maintenance process. The 3D modeling of these objects geometric information is parametric, in order to make the design more flexible. This research work offers an insight into the possibilities offered by different BIM-based tools for parametric modeling applied in the railway sector whereby an example of a railway section model is presented. Indeed, the focus will be on the creation of parametric objects representing railway components, as existing BIM object libraries lack them in the IFC2 × 3 standard format.

1. Introduction

Building information modeling (BIM) is defined as a process involving the generation and management of digital representation of the physical and functional characteristics of a facility [1], and it is now recognized as the most appropriate methodology to undertake building and infrastructure projects [2].

Since BIM has been mainly used in the building industry, the term I-BIM can be used when referring to infrastructure works [2].

Since the definition is recently coined, the terms "Civil Information Modeling" (CIM), "Heavy-BIM," and "Horizontal-BIM" [3] are sometimes used to distinguish it from the BIM for punctual works and vertical structures.

Based on the use of digital models to simulate the entire construction process and use throughout the entire lifecycle (planning, design, construction, and maintenance) of a built facility, the information flow between stakeholders involved at all stages is significantly improved. BIM results in an increase in efficiency by reducing the laborious and error-prone manual reentering of information that dominates conventional paper-based workflows [4].

Even if there have been efforts towards data interoperability with development of open-source standards and object-oriented models, such as Industry Foundation Classes (IFC) for vertical infrastructure, there is a huge lack of open data standards for the information exchange for horizontal infrastructure limiting the adoption and effectiveness of integrated BIM [5].

In this research work, a procedure for the creation of railways components' parametric object is presented, by investigating and comparing the use of different BIM-based tools, highlighting their potentialities/limitations.

2. Literature Review

Nevertheless, many efforts have been made in recent years to improve interoperability and to achieve better results in the integration of BIM in the infrastructure sector [6–11].

In the last years, there is a growing interest in the planning, design, and management of rail and metro transportation systems [12–17].

As far as the railway sector is concerned, the following is a brief review of experiences in several countries where I-BIM has been adopted.

In Germany [18], German Railways has considered the implementation of the BIM methodology for its railway projects with the aim of managing any BIM project of greater technical complexity by 2020. Specifically, an example of Deutsche Bahn's use of the BIM methodology is the excavation of a railway tunnel in Rastatt [19].

In Morocco [20], the Office National des Chemins de Fer (ONFC), national operator of the rail network, planned to build 1500 km of high-speed rail lines by 2035, 360 km of which were delivered in 2018. Taking in account that the COLAS Rail Morocco team that led this project consisted of 15 designers and engineers from different disciplines, the BIM integration speed turned out to differ from one discipline to another; while architectural and structural aspects were more likely to adopt the approach, other disciplines have encountered problems, especially related to interfaces with other dedicated software. The 3D modeling was made based on existing 2D plans; the 3D model was implemented by Revit® software, architecture module, which is close to the software (2D) used by the team in the past and which allows an optimal connectivity with other tools, especially for the calculation. The structure and networks modules were then used to design the civil engineering structure, the metal structure, and the electrical equipment.

In Norway [21], the Norwegian National Rail Administration has edited the "Manual for digital planning" which integrated 3D modeling, BIM, and infrastructure projects. In particular, the first major investment involved was the intercity rail project around Oslo, started in 2016. This was the country's largest investment in railway infrastructure in recent times in terms of billions of Euros. 3D/BIM initiative was considered a crucial requirement for the companies involved, resulting in significantly fewer errors and rework on the construction site, with substantial savings and better quality in all areas.

In Portugal [22], a 35 km-long section of railway infrastructure was rehabilitated in 2016 using a geotextile and geogrid in the ballast layer base. The application of the BIM methodology consisted of three phases: (1) creation of the project model (3D), (2) planning and scheduling of the construction process (4D), and (3) extraction of model information. The objectives of these phases were achieved in general using different software packages. The Civil 3D® and Revit® packages proved suitable for 3D modeling. The Navisworks® program was used to create the 4D model.

In Sweden [23], even in the absence of clear-cut government-led guidelines, the construction companies (especially the firms producing mostly single-family homes) and architect companies have pioneered the adoption of BIM since the mid-2000s. Engineering companies were open to new technologies and understood the benefits of BIM and the opportunities to develop services since 2007. As far as the infrastructure field is concerned, Sweden's largest transport project administration, the Swedish Transportation Administration, published a BIM strategy in 2013 with the aim of including BIM for all new investment projects from June 2015. All major design firms have begun to develop the use of BIM and its capabilities.

In Slovenia [24], as part of a European project for the rail interoperability of a line connecting Lyon and Budapest and beyond to the border with Ukraine, the BIM approach was used for a "reverse engineering" process to use the model generated for the future operation and maintenance of a road underpass to a railway near Grlava Station in the northeast of the country.

The BIM-based tools used are (1) Autodesk Civil 3D® for georeferencing the work and corresponding terrain modeling; (2) subassembly composer for cross-section modeling, carried out by programming a decision tree that relates geometric elements with the definition of constraints and rules to automate the development of the section; (3) Autodesk Revit® to create a new "family" of railroad tracks, sleepers, and docks; and (4) Allplan BIMplus®, a OpenBIM platform, for importing models in order to extract all the information.

In the UK [25], since the publishing of the GCS (Government Construction Strategy) in 2011, the UK government has already started implementing BIM on several early adopter projects. Projects of interest are (1) Crossrail, under central London to link existing network rail lines to the east and west of the city [26]; (2) High Speed 2 (HS2), implementing a high-speed network from Glasgow, Edinburgh, and Newcastle to London, converting existing lines and integrating them with others to be built [27, 28]; and (3) the redevelopment of about 350 km of the Northern Hub Rail [29].

In South Korea, the 9.38 km Honam high-speed line, including 2 tunnels and 6 bridges [30], includes the use of BIM.

Kenley et al. [1] presented the BIM projects for the new 12.6 km Moreton Bay Rail high-capacity line in Australia and the extension of the Kuala Lumpur Light Rail Transit line by 35.5 km in Malaysia. Critical issues emerge in parametric modeling and data transfer from CAD to GIS related to poor interoperability, hence the need for research towards the development of open data formats.

In France, in 2014, the Ministry of Housing and Construction declared that the use of BIM would become mandatory in public markets from 2017 [23]. It should be noted that "BIM France" (association of architects and engineers) had already decided to actively support the development of BIM in France one year earlier.

Another main actor in this scenario is Bouygues immobilier whose CEO, François Bertière, adopted the "Central BIM Policy." In this document, the construction company describes in a precise way its vision of the BIM and the way in which it will proceed to its generalization by 2020

with the goal "to ensure the design and implementation of all works under integrated BIM (the highest level of BIM) by 2020" [31].

In the railway sector, SNCF Réseau (railway infrastructure manager in France) is working on its new projects (such as CDG Express and Grand Paris) by integrating BIM. In addition to the creation of the BIM 3D model, the goal is to also implement the 4D to simulate construction work (phasing) [32].

2.1. Aims and Methods. The research study aims to investigate on the potentialities in using different BIM-based programming tools for applications in the railway sector, testing the capabilities of interoperability and information exchange between them.

The investigated BIM tools (Figure 1) are based on different scripting languages for parametric modeling, which allow greater flexibility in the process of model parameterization.

Then, the interoperability between BIM tools has been tested using a BIM viewer.

Therefore, not only the advantages derived from the adoption of these tools have been highlighted but also their limits have been explored, mainly due to their early stage of development and the reliability on the designer's skills for the success of the process.

2.2. Case Study. As a case study, the process for modeling a single-track straight-axis railway section is presented.

For this purpose, OpenSCAD®, Rhinoceros®, and Dynamo® design-tools that allow the visualization and parameterization of the model were used.

OpenSCAD® is a free software for creating solid 3D CAD models using his own coding language, such as C++, the designer can directly modify the code on which the model and its graphical representation are based.

Rhinoceros®, often called just Rhino6, has a different approach; it takes CAD-based methodology to the next level working along with several plugins, as the well-known Grasshopper®, based on flow-based programming graphs made by code-logic elements connected on a canvas and the BIM-based tool VisualARQ®.

Dynamo® is also a flow-based programming platform with a visual interface, and this allows users to construct logical routines that automate workflows and help to save time.

The graphic rendering of the work is made possible by the internal rendering-tools of the abovementioned software, but for the rendering of BIM models, BIMcollab ZOOM was used.

The structure of the code has been designed to make it possible to model the structural element in accordance with the Italian and Slovenian Railway Design Standards, in particular:

(i) Istruzione tecnica RFI TCAR ST AR 01 001 A del 26/07/2006 "Norme tecniche per la progettazione dei tracciati ferroviari" (technical instruction RFI TCAR ST AR 01 001 A of 26/07/2006 "Technical standards for the design of railway tracks") [33]

(ii) "Pravilnik o zgornjem ustroju železniških prog" (rules for the upper structure of railway lines) [34]

2.3. OpenSCAD Procedure. The code is composed of objects, actions, and operators. Objects are the building blocks for models, created by 2D and 3D primitives. Actions statements include objects creation using primitives and assigning values to variables. Operators modify the location, color, and other properties of the objects, using braces "{}" when their scope covers more than one action.

In the first part of the code (Figure 2(a)), all the variables in use have been defined; they represent the parameters of the model, allowing to modify the output geometry to the user's liking, as well as to respect the reference legislative sources.

The code is structured in modules (Figure 2(b)), i.e., subparts of the code: conceptually, each module defines a different part of the railway structure, i.e., base, ballast, plant layer, rails, anchorage system, and sleepers. A module is encoded with the name chosen by the designer, preceded by the keyword "module," and followed by the list in the brackets of the variables used within it ⟶ [...], followed by the list of arguments. The "basecore" and "head" modules (Figure 2(c)) contain the polylines used to create the 2D section of a single rail. They are then called in the "rail" module for the following linear extrusion to create the 3D object of the rail (Figure 2(d)); in the "rail" module, the operator "color" is used to give a "silver" color to the rail.

In the "ties" module (Figure 2(e)), a basic "cube" element was constructed and then moved to the desired position, and cyclically, this positioning operation was repeated along the length of the work with the "for" operator, an operator that makes it possible to iterate a command under certain conditions, i.e., starting point, interval, and final value at which the iteration stops. The preview of the ties is shown in Figure 2(f).

The modules "ballast," "track_ballast," and "grass" have a similar structure as they consist of a polyline for the 2D section which is then extruded linearly to create the 3D object of the railroad.

Three modules have been created for the bolts anchoring the sleepers to the railway structure (Figure 2(g)): "bolt" for the single bolt, "line_bolts" for the single row of bolts, and "bolts" for all the bolts. In the "bolt" module, the single bolt is created from primitive elements, such as the cube for the base plate and the cylinder for the central pin (Figure 2(h)). This module is called up in the "line_bolts" module to create the first pair of bolts. In the same way, the "line_bolts" module is called up in the "bolts" module in a "for loop."

3. Rhinoceros 6 with Grasshopper Procedure

Rhinoceros® is a particularly versatile software because of its multidisciplinary application from construction to mechanics to specific detail design. It turned out to be very advantageous to our purpose for its Grasshopper® plugin,
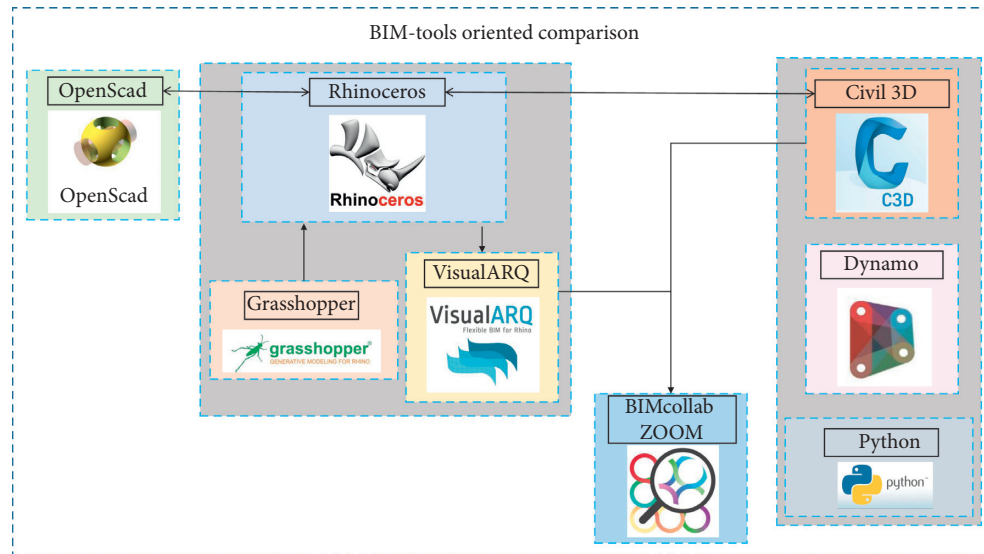
Figure 1: Methodology schema.

which allows designing parameterized objects using generative algorithms built with visual programming language that lets users create programs by manipulating elements graphically rather than by specifying them textually.

Grasshopper® is a visual programming language (VPL) environment-based software that uses the idea of "boxes and arrows" where boxes are the entities connected by the arrows that represent relations. The components that make up the code have been grouped together in different groups highlighted in colored squares.

First, an input group (Figure 3) was created in which the variables were divided into two main subgroups: in one, the variables are defined according to the Italian Geometric Design Standards and in the other, according to the Slovenian ones. The two groups of variables are then filtered by the "filter" component, and from this, the information flow passes to a third subgroup where the variables are listed again with their final values; so, depending on which geometric design standards is chosen, through the "filter," the variables will automatically assume the set of values previously defined in one of the other two subgroups. Two sets of inputs, not affected by the geometric design standards, are related to the plant layer, composed of soil and grass, and the total length of the modeled section.

The "filter" component has been used every time the designer has to choose between two or more options; it is a simple direct way to transpose an "if" construct into this graph logic. The filter takes an integer number as a value associated with the choice of the parameter set. The output of the filter is a group of values that are enumerated following the order in which they were entered into the filter component. Therefore, with the component "list item," every single element of the list must be extracted to be visualized and then used.

This process is necessary for the designer to choose the reference rails standard (Figure 4(a)), i.e., FS 46, UIC 50, UIC 60, and UIC 71, as far as the Italian Standard is concerned, and S 49, UIC 54E, and UIC 60, as far as the

Slovenian Standard is concerned; the type of railway (Figure 4(b)), i.e., A (major) or B (secondary), is according to Italian Standard, National, Regional, Industrial, according to the Slovenian one, and the subballast layer material, i.e., concrete or asphalt; the construction material for the ties (Figure 4(c)) is wood, iron, or concrete. Each variation of these inputs determines a different geometry of the elements in the structure.

Other inputs are the gauge, i.e., the distance between the rails, that can be usually used in Europe (143.5 cm) or reduced (95 cm in Italy and 76 cm in Slovenia); the distance between the outer rail and the shoulder of the ballast; the distance between the ballast shoulder and the subballast shoulder; the inclination of the ballast and subballast layers; the thickness of the base; and the transverse inclination for the runoff of the rainwater.

After all parameters have been defined, the designer chooses the reference standard (Figure 4(d)).

Each part of the railway body, except for the crossbars and the anchorage system, has been modeled by extruding along a common direction, that of the railway axis, a surface defined from polylines whose vertexes have been defined before.

Since they are 2D surfaces, the vertices to be defined are points with only two coordinates, "x" and "z" according to the reference system, while "y" is the axis of the railway axis.

Each coordinate calculated using the "format," renamed in "X_3rd Point_Sub-Ballast," has as input in "F," the output of the "panel" (yellow square) that is the mathematical formula that manipulates the other inputs of the "format" (Figure 5(a)).

Coordinates were enlisted in "list of X coordinates" and "list of Z coordinates" and then combined to define the points that have been used by the "polyline" element to form closed polylines and then used to create the surfaces, through the "boundary" element (Figure 5(b)).

The "polyline" element (Pline) allows to create closed polylines using as input a vector of points. The "boundary"
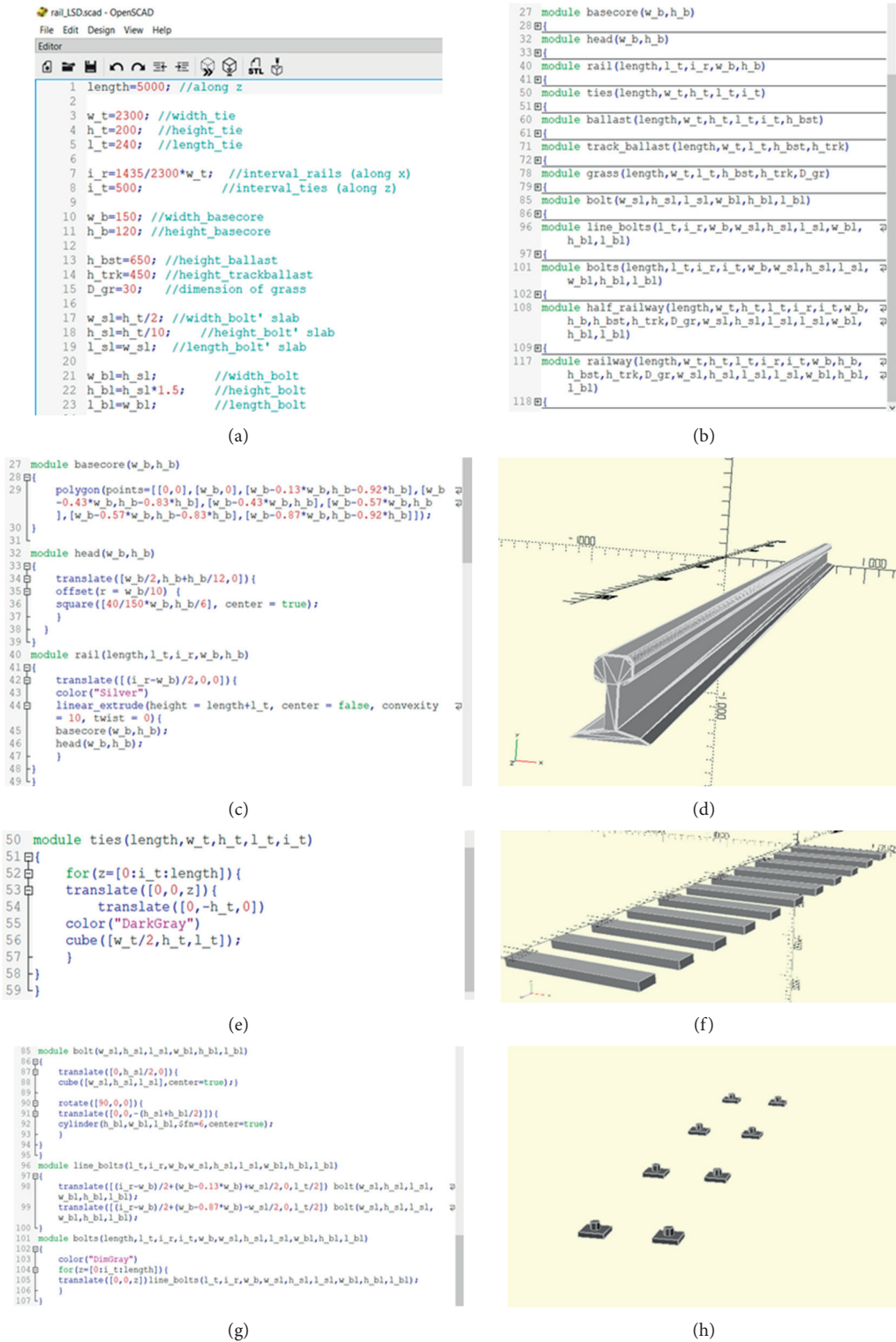
Figure 2: Representations of some common OpenSCAD output. (a) Definition of variables involved. (b) Modules. (c) Code for the rail representation. (d) Single rail preview. (e) Module "ties." (f) Ties preview. (g) Code for the bolt representation: modules "bolt," "line bolts," and "bolts." (h) Bolts preview.

element allows to create the surface whose boundaries are formed by the polylines.

Figure 6(a) shows a preview of the basic section. After the extrusion and adding information about color to each element, it is possible to generate a preview of the complete extruded section as shown in Figure 6(b).

To obtain the crossbars, initially, it was thought to create a single crossbar and then use an iterative process that would
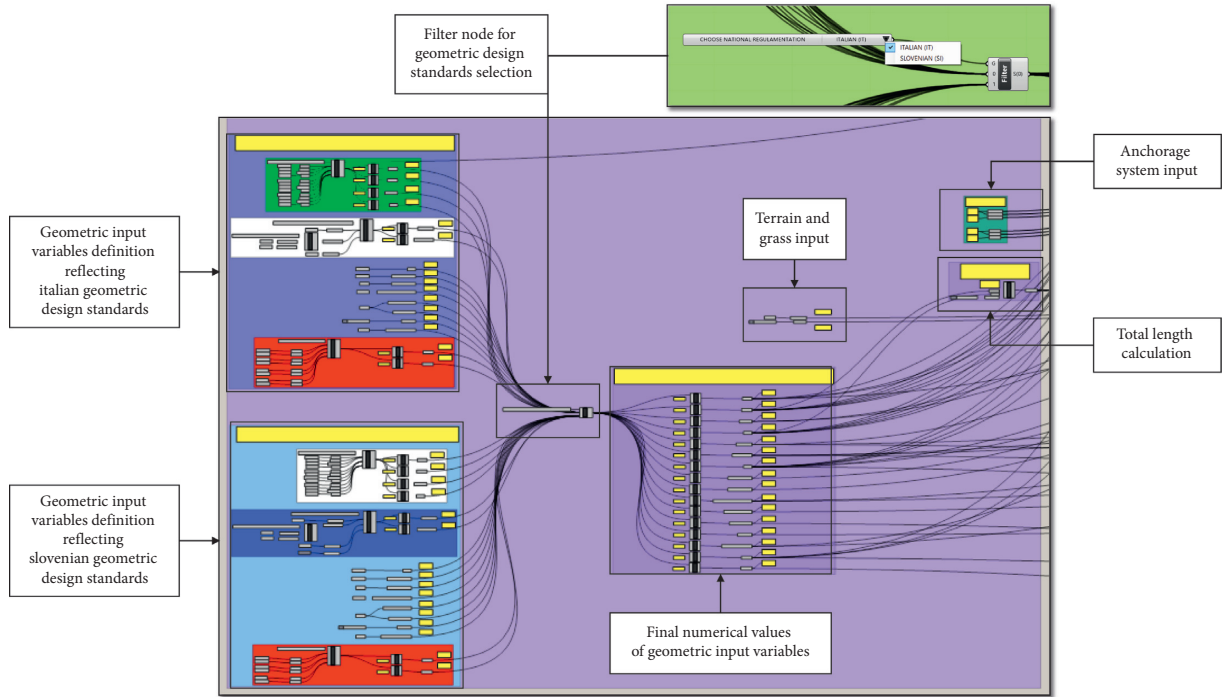
FIGURE 3: Input variables group.

(a)

(b)

(c)

(d)
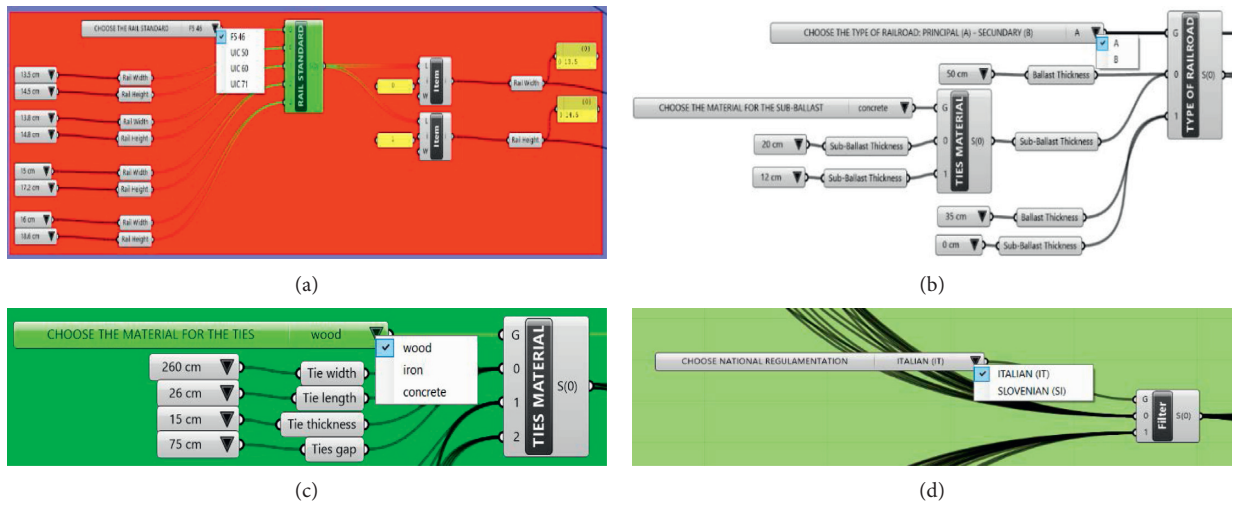
FIGURE 4: Use of the "filter" node. (a) "Filter" used to choose the rails standard. (b) "Filter" used to choose the type of railway and the material of the subballast. (c) "Filter" used to choose the material of the ties. (d) "Filter" used to choose the standards of reference.
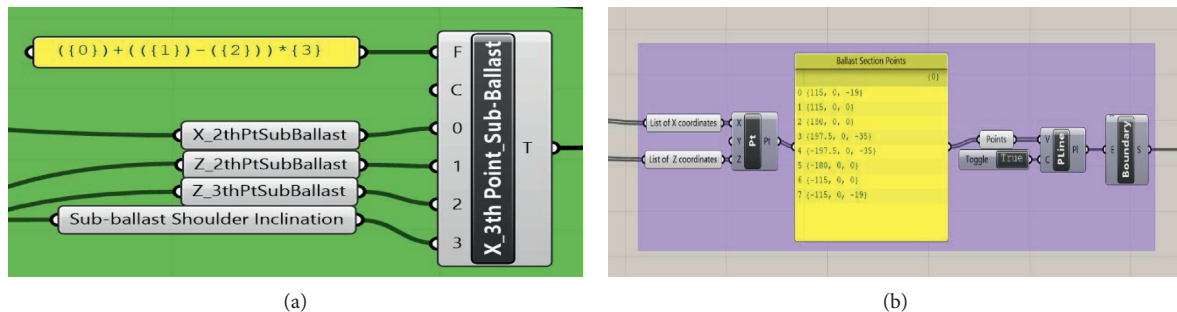

(a)

(b)

FIGURE 5: Points, pline, and boundary definition. (a) Format element. (b) Points element.
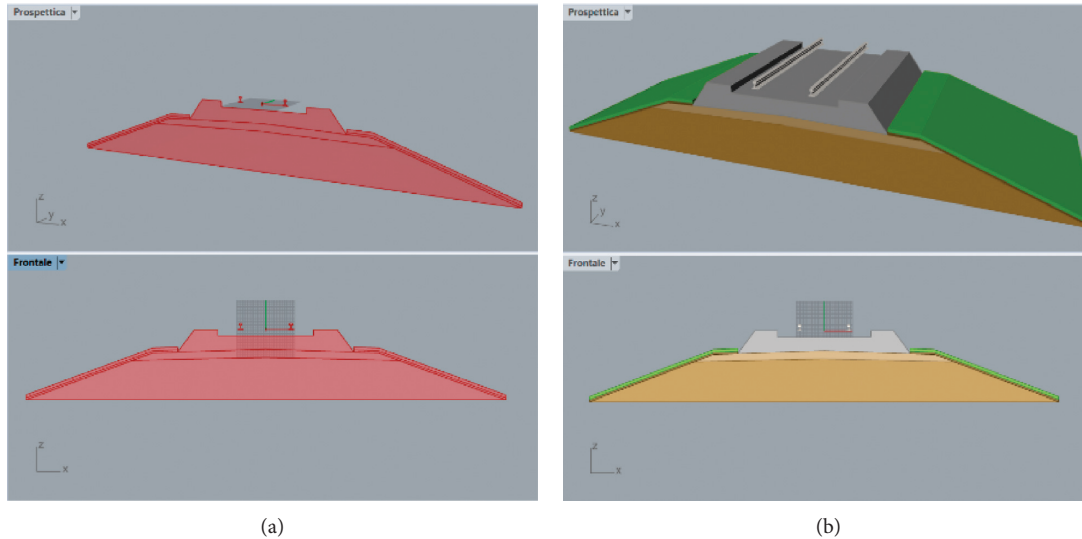
(a)   (b)

FIGURE 6: From 2D to 3D model. (a) Preview of the section. (b) Preview of the extruded section.

create a new crossbar in the same position as the first one, overlapping them, and then moving it to the next position.

This process, in the classic setting of coding, for example, in OpenSCAD®, would be coded with a "for loop" in which the initial position was the origin of the reference system, and the interval was the distance between two successive crossbeams and the end was the total length of the structure. To use the "for loop" in Grasshopper®, it is possible to use Python language code written inside a "panel." In this work, we have opted for another solution: in a nutshell, a series of rectangles were produced in the plane $[x, y]$ using an iterative process and then extruded along z to create the solids-crossbars (Figure 7).

The iterative process to build the crossbeams is made up with three "series" elements.

The first two have input the sizes of the top of the single crossbar, and they produce a certain number of squares in the plane $[x, y]$ depending on how many times the element must be repeated depending on the length of the railway.

To define how many times the element must be repeated, the third "series" element was used that operates with the total length of structure and with the gap between two consecutives cross beams as input.

Finally, the 2D geometry has been extruded for the height of the cross beams along the z-axis. The same pattern was used for the anchorage system: first was constructed the 2D geometry on plane $[x, y]$ and then extruded along z. The preview of the entire model is shown in Figure 8.

4. Civil 3D® with Dynamo Procedure

Dynamo® visual interface is easier to use for those who are not experts in coding, since it is a visual programming tool based on graphs made by nodes that represent a single line of code and arrows that connect these nodes implementing logical structures. It is a VPL environment software as Grasshopper®.

The workspace (Figure 9(a)) contains standard menus (File-Edit-View) and toolbars. Dynamo® contains an extensive library of nodes and commands, on the left, that are organized by category such as Geometry, AutoCAD®, and Civil 3D®. On the left bottom, there is the execution bar which is set by default to automatic execution, but can be changed to manual, and this is a suggested action in case the workflow becomes more complex and consequently, the execution becomes more onerous.

Nodes are classified by three behaviors (Figure 9(b)): the green plus represents nodes that create an element, the red lightning bolt represents an action method, and the blue question mark represents an input or property query. When a user expands the node tree and hover over a node, it presents the inputs and outputs for that node. Users can also search nodes by name in the search field: any node with the inputted information appears on that list.

In this case study, Dynamo® was mainly used for the creation of the geometric representation of some parts of a railway section, in the context of the use of Civil 3D® for the railway design.

Starting from a topographic map, the digital terrain model (DTM) was generated [35]; then, a single-track railway alignment was created, and finally, the corridor was modeled by extruding the standard section type available in the catalog library. Figure 10 shows the corridor graphical output, which consists of a series of "baselines," i.e., extrusion guidelines and "feature lines" that connect the coded points that make up the vertices of the chosen section type.

Civil 3D® looks fast and versatile, as it allows to import also topographic maps created from surveys and investigations onsite; in addition, having integrated the Italian Geometric Design Standards, it permits to automatically generate graphs such as the elevation profile and the speed chart. However, it is not particularly accurate in the geometric modeling of the corridor, since it simply extrudes section type; so, as shown in Figure 10, the cross beams are
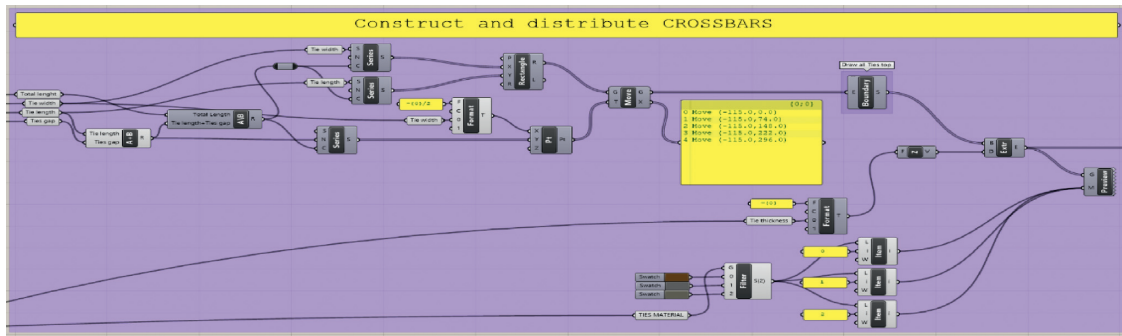
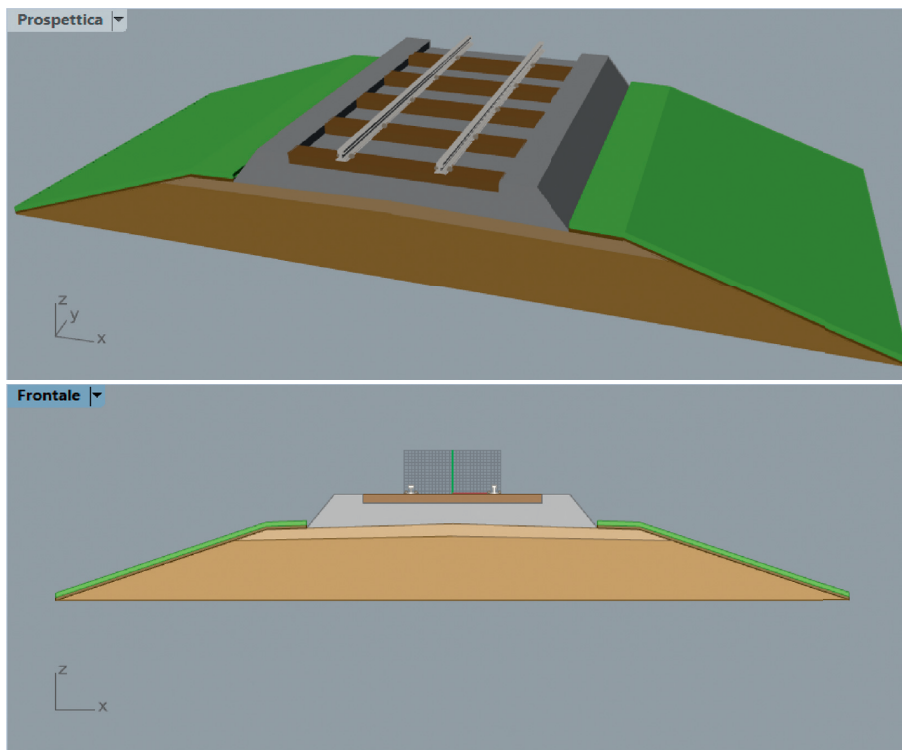Figure 7: Grasshopper code for crossbars.

Figure 8: Preview of the complete model.

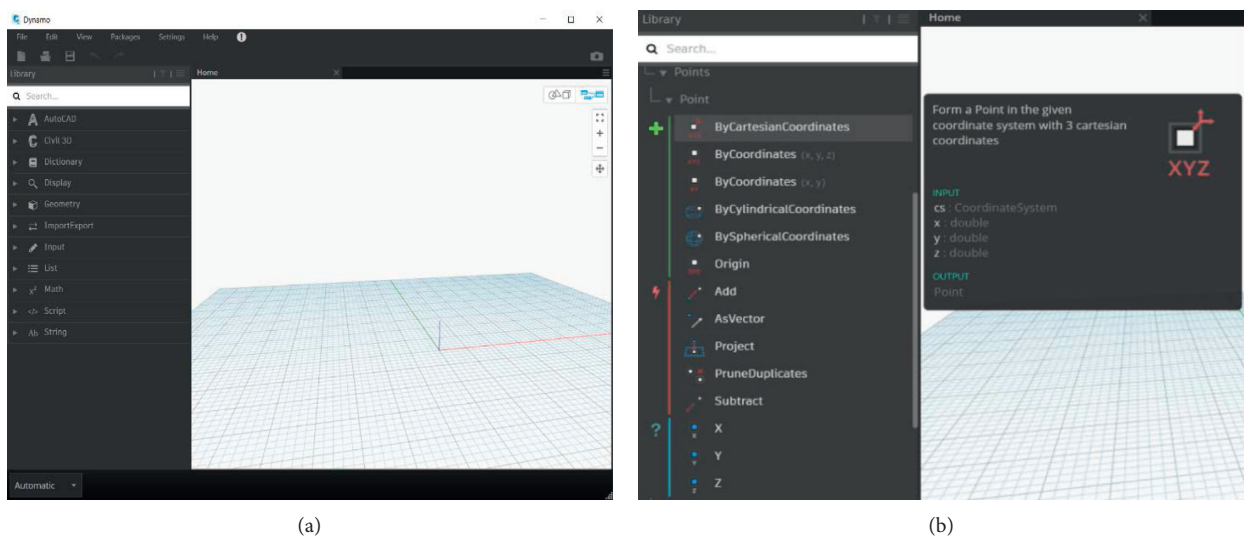(a)   (b)

Figure 9: Introduction to Dynamo®. (a) Workspace. (b) Nodes: classified by three different behaviors—type description of a node.
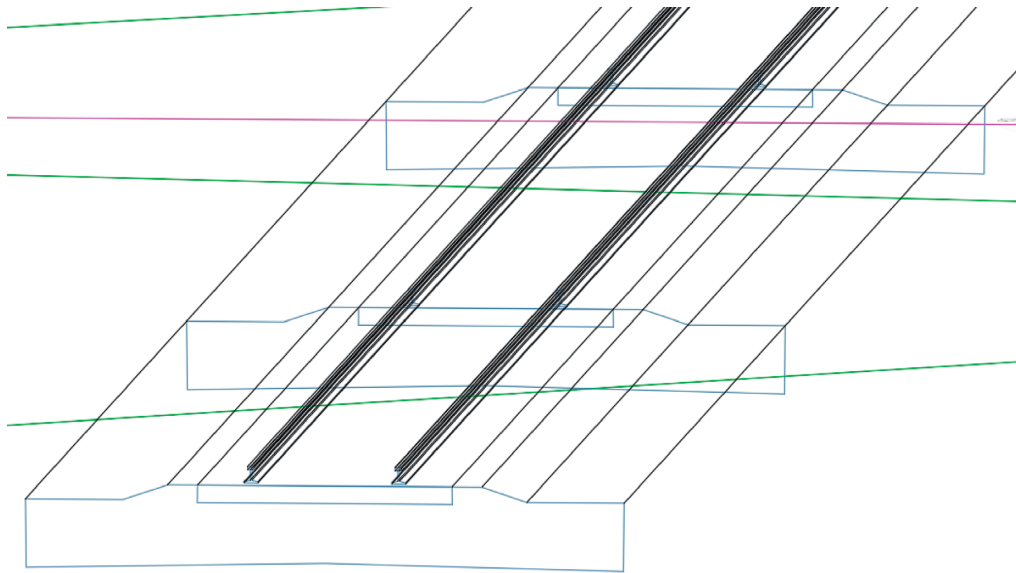
FIGURE 10: 3D wireframe view of the railway corridor.

represented as a single long extruded plate along the entire alignment.

Therefore, the procedure has been chosen as follows:

(i) The "Tie" object was modeled from the start in Dynamo®, imported in Civil 3D® as a block, and then distributed along the corridor in the right position to complete the corridor geometric model

(ii) Then, the section type has been modified to eliminate the object "Tie" of the section type. In this way, there is no interpenetration.

It is important to note that it has been possible to associate different information to the "Tie" object, such as information concerning geometric dimensions and material properties.

As far as Dynamo® application is concerned, it has been used first for the creation of the geometric object representing the single railway crossbar (Figure 11): the "Cuboid.ByLenghts" node allows the user to create a cuboid from its three dimensions, i.e., width (x), length (y), and height (z); these inputs were taken from a CSV file, i.e., an Excel-based file, but where the input is a text with values separated via comma, the output of this node goes into the "geometry" input gate of the "Object.ByGeometry" node, which allows to transform any geometry into a Civil 3D® object having previously indicated the layer to which it belongs (named "01. Tie") and a reference CAD block (named "01. TIE").

Associating a property set to an object is possible; but first, the property set should be defined (Figure 12(a)); in our case, geometric information and material have been added (Figures 12(b)–12(e)).

The same operation can be performed with a Dynamo script (Figure 13), creating a property set by object and updating one by one the properties of the set.

The next step was to distribute the 01. TIE block along the entire corridor.

In this experience, the block was defined in a file apart; anyway, it is possible to import any object or block defined in other files: open the chart in which the corridor is drawn; then, it is necessary to import the block (Figure 14).

To distribute the ties along the whole corridor, the Dynamo script shown in Figure 15 was used.

Figure 15(a) shows the first part of the script, where

(i) Through the node "Document.Current," the user selects the current document to which Dynamo® refers, that is the Civil 3D® file where the drawing of the corridor is present

(ii) The "Corridors" node extracts from the current document set of corridors represented in this case that there is only one corridor

(iii) The "Corridor.Baselines" node extracts the set of all the baselines that make up the corridor

(iv) Through a "Code Block," it is possible to extract from the list of baselines the one we are interested in

(v) The node "Baseline.CorridorFeatureLines" allows the user to extract the list of feature lines that connect the coded points making up the type sections (subassembly)

(vi) Through a second "Code Block," the user can extract the feature line of interest, that is, the one along which to distribute the single tie

Figure 15(b) shows the second part of the script, in which

(i) Starting from the "Code Block" of which before, through the nodes "CorridorFeatureLine.Start-Station" and "CorridorFeatureLine.EndStation," the "start" and "end" stations of the feature line are selected

(ii) With the node "Range," the user creates a sequence of points along the feature line from the start station to the end station with a step defined by a
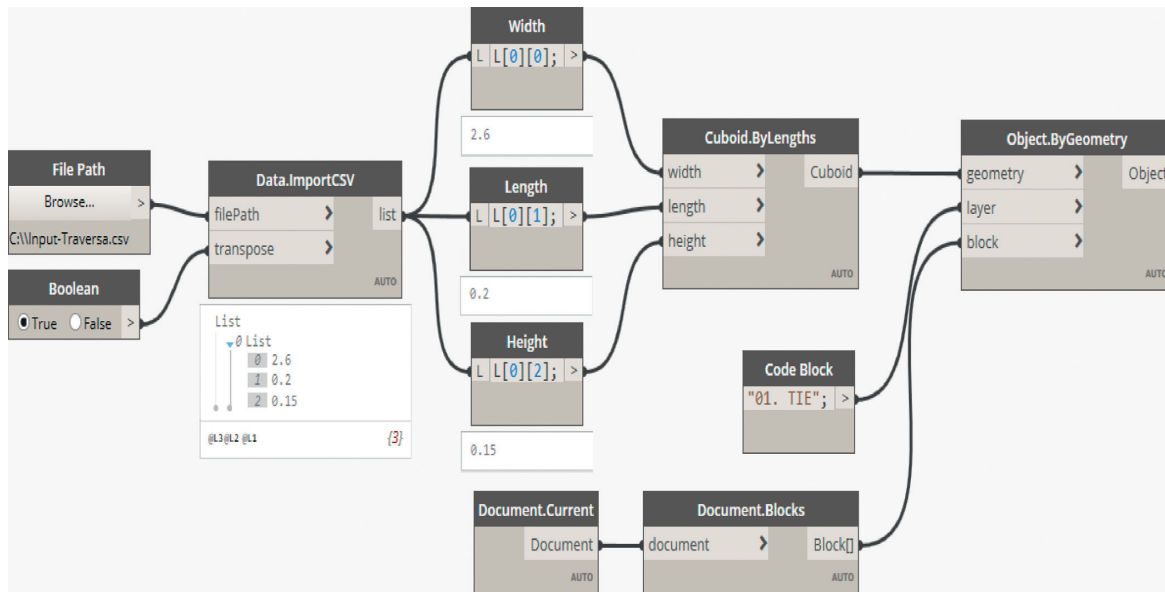
FIGURE 11: Script for the creation of the geometric object of the single tie.

Python script that defines the gap between two successive ties according to the type of railway (primary or secondary because of different speeds and weights of trains) and the type of section (whether in line or in a crossing because of different speeds of trains)

(iii) The "CorridorFeatureLine.CoordinateSystemBy Station" node creates as many as three-axis coordinate systems, X: Y: Z, as defined by the range along the feature line

(iv) Of these coordinate systems, it is good to isolate the origin and the Y and Z axes via the nodes "CoordinateSystem.Origin," "CoordinateSystem.YAxis," and "CoordinateSystem.ZAxis"

Figure 16 shows the third and last part of the script, in which the block of objects that will constitute the distribution of the traverses along the path is created. The key node is "BlockReference.Create," the node that creates the block.

It has input

(i) A sourceBlock provided through the output of the node "Document.BlockByName" which allows the user to select a block from a document (in this case, the current document is obtained through the node "Document.Current") and the name of the block itself previously created in Civil 3D® (in this case, provided through the node "String")

(ii) A position, which in this case is actually a list of positions, i.e., the list of origins of previously created reference systems

(iii) A normal reference position, i.e., the Y-axis of the previously created reference systems

(iv) A layer to be referred to in Civil 3D®, always indicated by a "String" node

(v) And the block from model space indicated by the node "Document.ModelSpace"

5. Results

As far as OpenSCAD® application is concerned, once all the individual component parts have been created, they are called inside the "half_railway" module and combined to create half of the railway. Then, in "railway," the second half, next to the first, is created with the "mirror" operator.

The result preview is shown in Figure 16.

As far as Rhinoceros® and Grasshopper® application is concerned, once the geometric modeling was finished, the last and very important operation was the "baking." Through the "Geo" component, it is possible to define the geometric elements built in Grasshopper® as objects that can be modified in Rhinoceros® (Figure 17(a)) with addition of semantic information and IFC export options (Figure 17(b)).

Once the modeled objects have been assigned features in Rhinoceros®, it is possible to save the file in IFC3x2 format, thanks to the VisualARQ® plugin. It is then possible to open the file with the BIMcollab® software.

BIMcollab® immediately shows a compatibility problem because the software does not display the selected material in Rhinoceros® but only receives the color assigned to the layer. In fact, the software does not read the characteristics of the material, and the field related to it in the "component properties group" remains empty.

To solve the problem, the IFC 3×2 file was manually modified in a simple text editor. As far as Civil 3D® and Dynamo® application is concerned, the process of transforming the geometrical model into a semantic oriented object was automatic and continuous during all the design procedures.
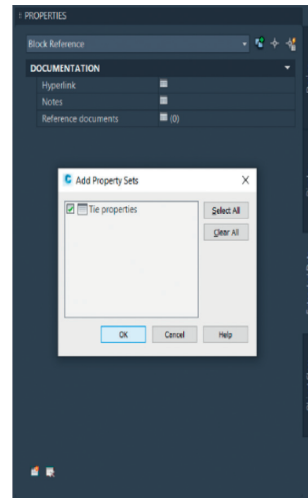
The modeling output result in the output view is shown in Figure 18.
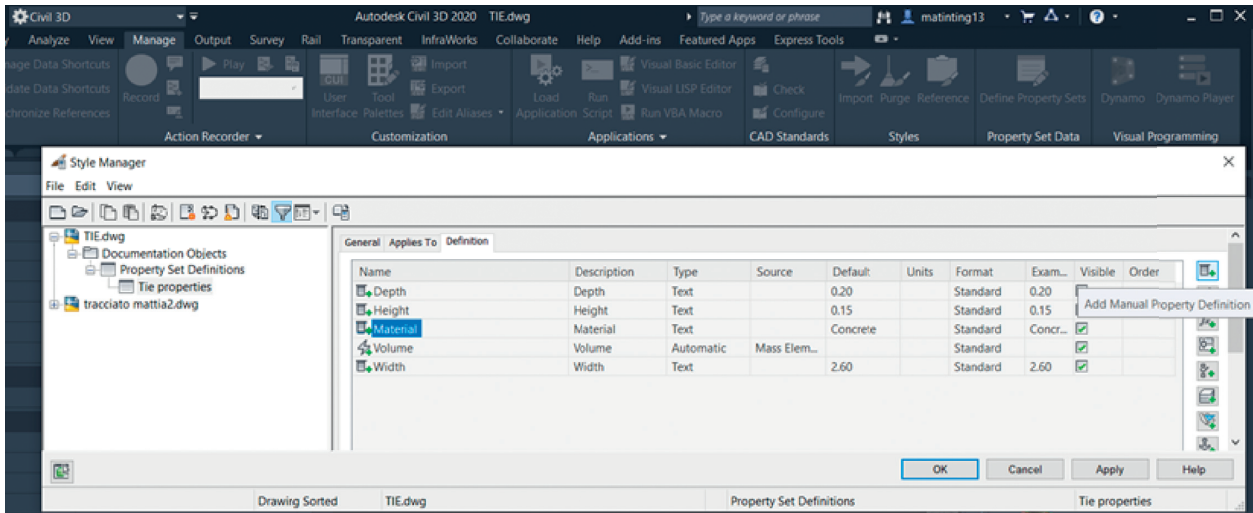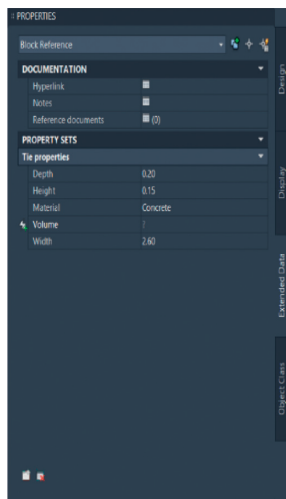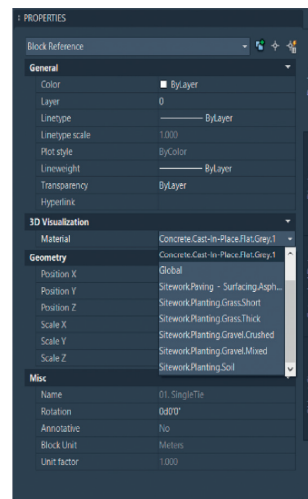
(a)
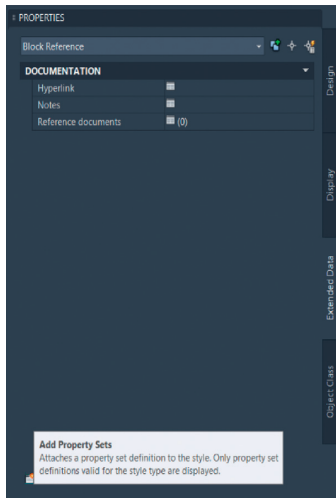
(b)

(c)

(d)

(e)

FIGURE 12: Adding a property set to a block of objects in Civil 3D®. (a) Property set definition. (b) Add property sets. (c) Select the property set in the property sets list. (d) Visualize and personalize data. (e) Adding a default material in properties—design.
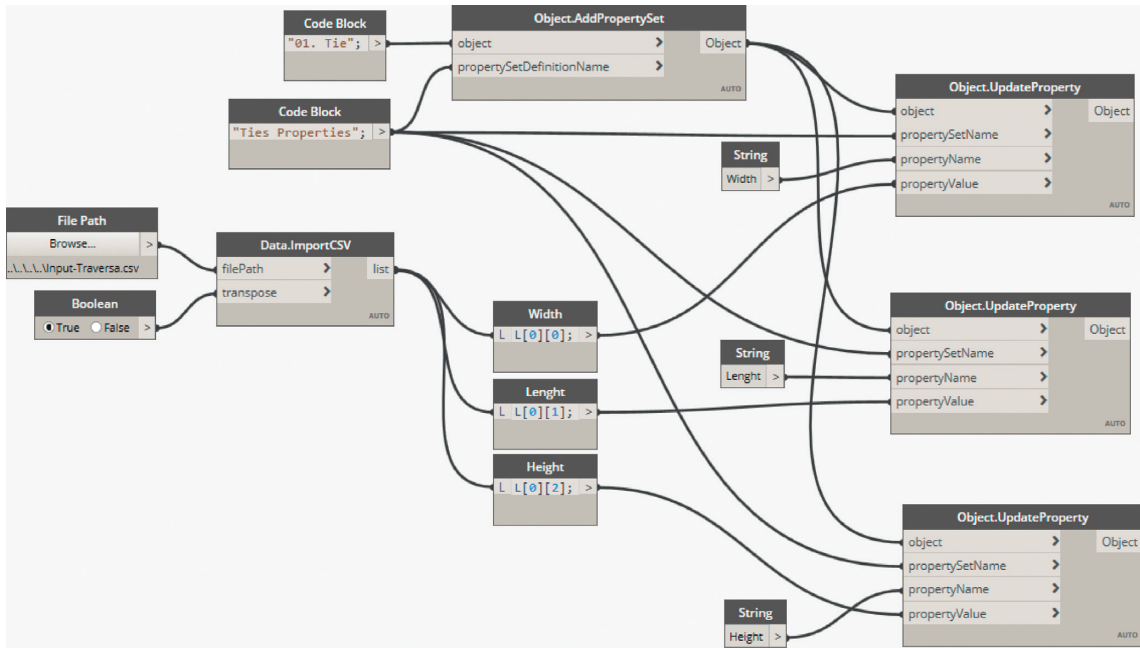
Figure 13: Dynamo script let the users setup and update a property set to a defined object.
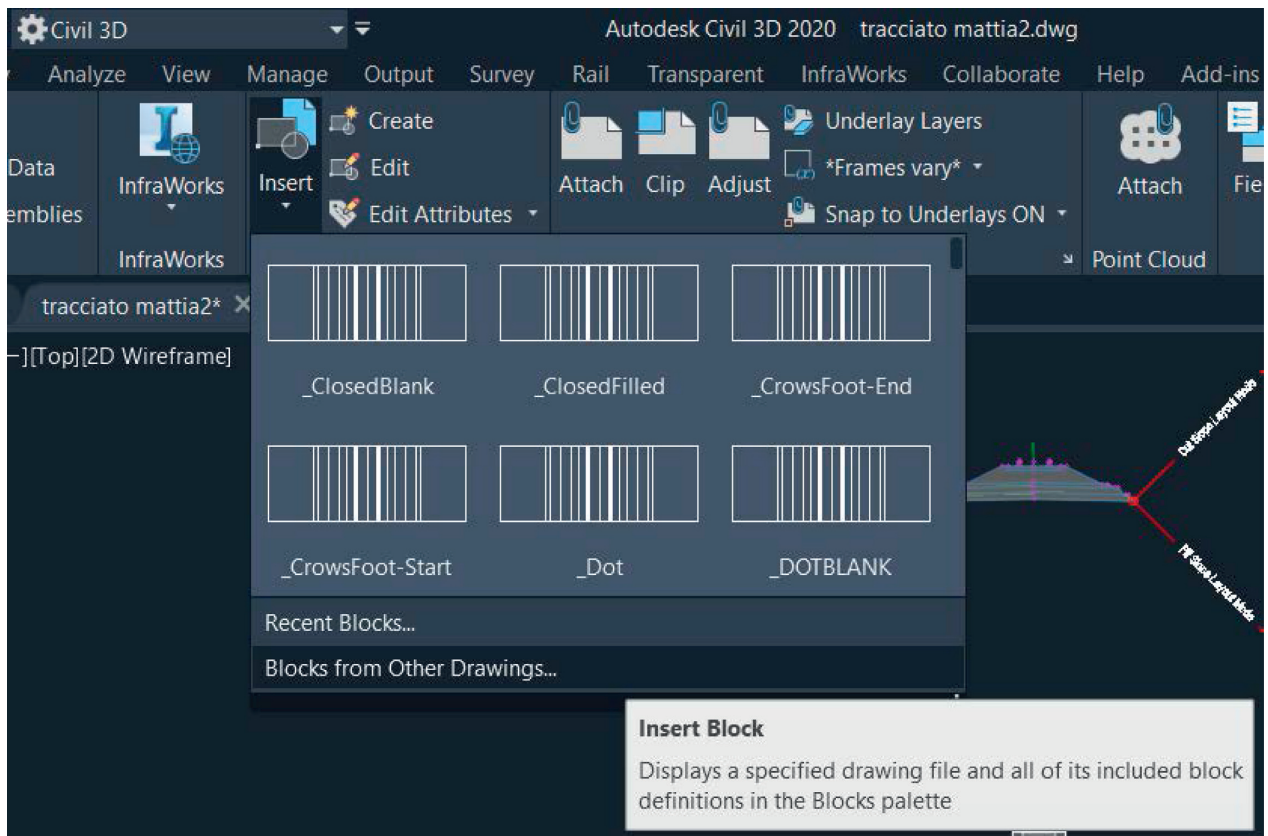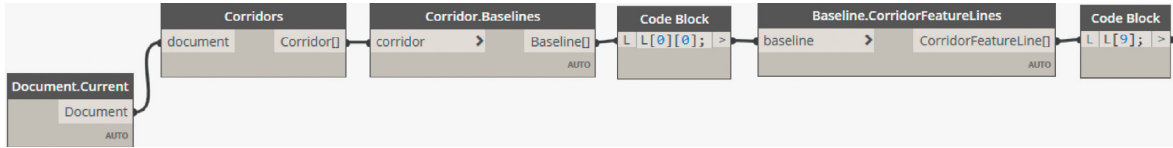

Figure 14: Importing a block from other drawings.

(a)


```
30
31
32 with adoc.LockDocument():
33     with adoc.Database as db:
34
35         with db.TransactionManager.StartTransaction() as t:
36             # Place your code below
37             RailwayType=IN[0]
38             RailwaySectionType=IN[1]
39
40             if RailwayType == "Main":
41                 if RailwaySectionType == "Line":
42                     gap = 0.60
43                 if RailwaySectionType == "Crossing":
44                     gap = 0.667
45             elif RailwayType == "Crossing":
46                 gap = 0.75#
47             #
48
49             # Commit before end transaction
50             #t.Commit()
51             pass
52
53 # Assign your output to the OUT variable.
54 OUT = gap
55
56
```
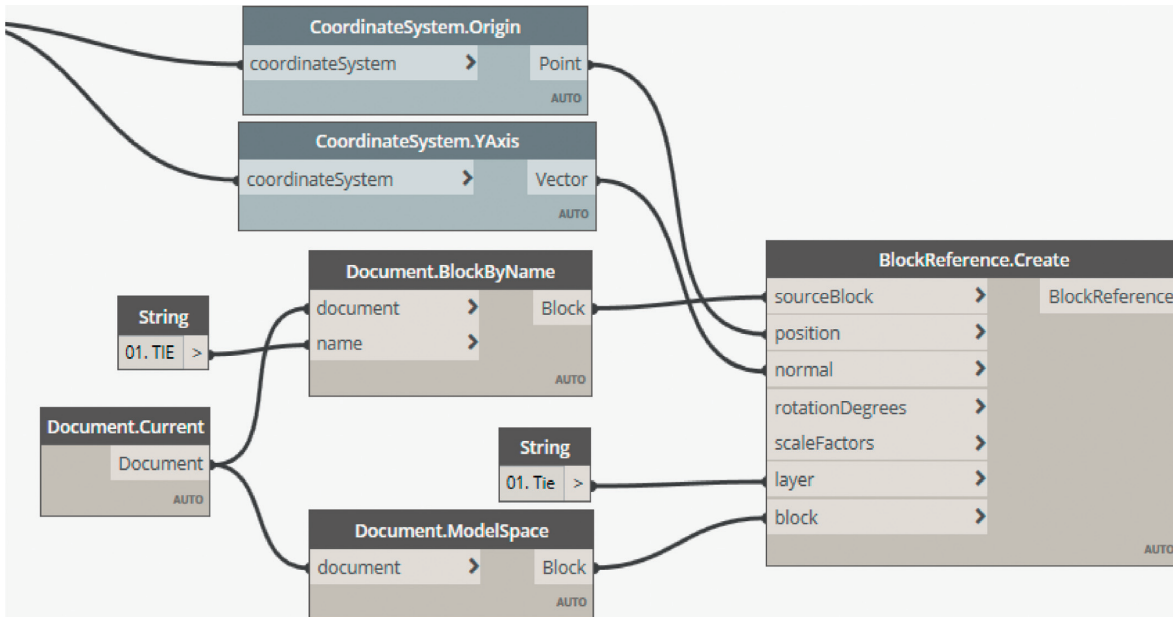
(b)

(c)

FIGURE 15: Script to distribute the ties along the corridor. (a) Reference feature line selection. (b) Selection of reference sections along the chosen feature line with Python Script aid. (c) The objects block creation.
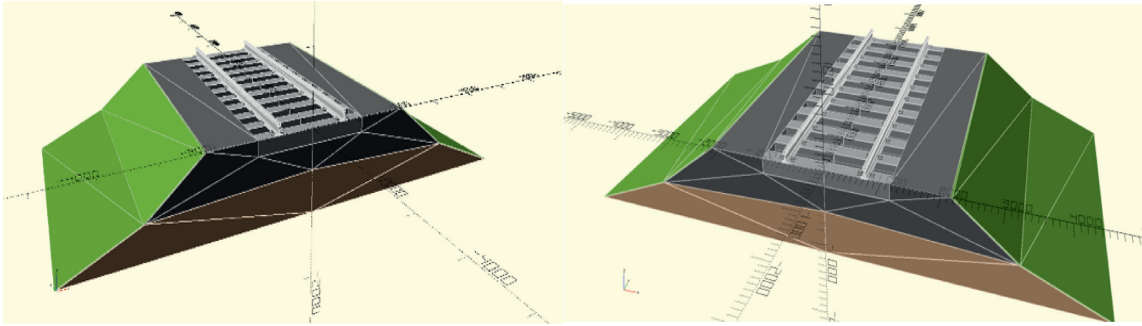
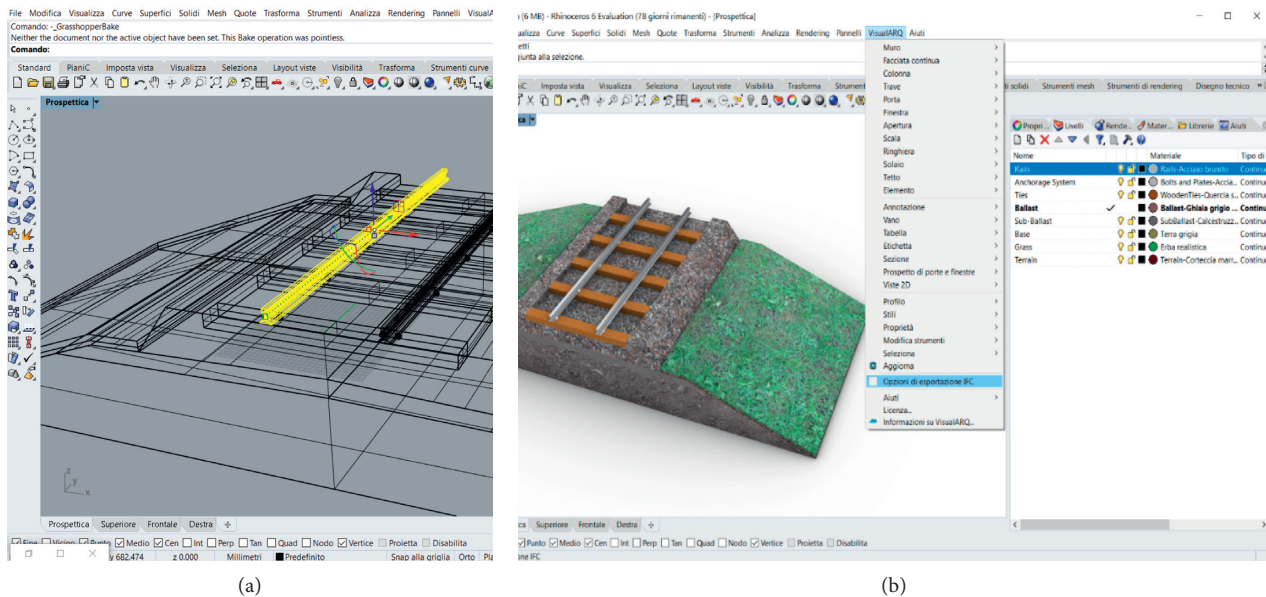FIGURE 16: Model displayed in different perspective views.

(a)

(b)

FIGURE 17: Rhinoceros. (a) Objects. (b) Semantic information menu and IFC export options.

6. Discussion

This study has shown how different BIM-based tools can lead to satisfactory results for different needs.

   (i) OpenSCAD® allows a dynamic visualization of what is produced by the code, but it is limited to geometric modeling;

   (ii) Rhinoceros® offers a very intuitive interface, uses NURBS for explicit modeling and Grasshopper® for implicit modeling, but it is not an infrastructural design software, and this leads to limit both geometric and information content. To export to IFC, it uses VisualARQ®, but this uses only architectural standards and limits IFC to the version 3×2;

   (iii) Civil 3D® is the most specialized tool for an infrastructure project, being specifically designed for this field and implementing the geometric design standards. Dynamo® is a tool with great potential for the automation of model production.

All BIM-based tools are supported by the main operating systems.

As shown in Table 1, OpenSCAD® is the only one open source and is not using plugin.

As for the file formats used, the list is very long:

   (i) OpenSCAD® can export 2D drawings in the DXF format and 3D models in STL, both formats recognized by Rhinoceros® but not by Civil 3D®

   (ii) DWG and IFC are instead the interchange formats between Rhinoceros® and Civil 3D®

Regarding the programming characteristics,

   (i) OpenSCAD® has a traditional but intuitive code language, which implements BREP and CSG techniques by efficient modeling of solid forms. Being able to dynamically visualize what is created and how the geometry changes according to the input parameters is a great help for the designer who traditionally sees the result of his code only when it is finished.

   (ii) Grasshopper® and Dynamo® are both based on visual programming language. Grasshopper® has a much larger library of logical nodes, allowing the
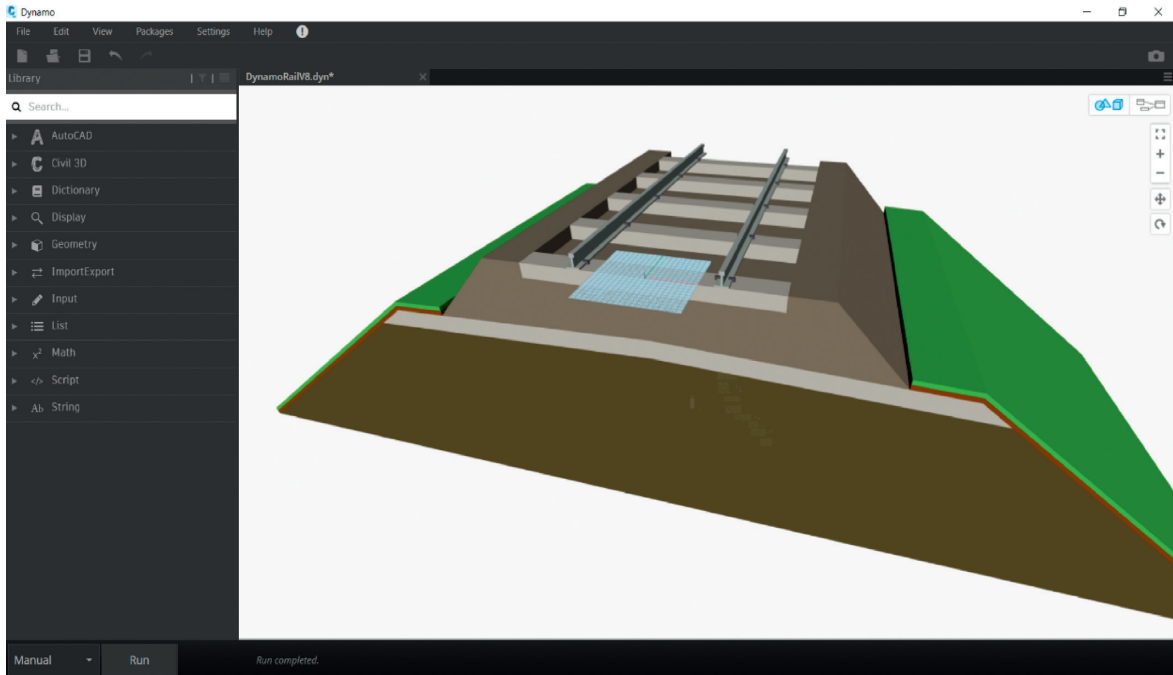
Figure 18: Model in Dynamo® for Civil 3D® output view.

Table 1: BIM-based tools comparison.

	OpenSCAD®	Rhinoceros®		Civil 3D®
Plugin	—	✓		✓
		Grasshopper	VisualARQ	Dynamo
License	Open source	Proprietary software		Proprietary software
Operating System				
Microsoft Windows	✓	✓		✓
MacOS	✓	✓		✓
Linux	✓	✓		✓
File format exchange				
STL	✓	✓		—
DXF	✓	✓		—
DWG	—	✓		✓
IFC	—	✓		✓

user to compile, in general, more elaborate and complex algorithms than Dynamo®. The good thing about Dynamo® is that it is a Civil 3D® plugin, but it can also work on its own, via the Dynamo Studio app. In addition, the model is displayed on the same screen as the Canvas, i.e., the VPL graph display window. In contrast, Grasshopper® also depends directly on Rhinoceros® for display.

Therefore, also some problems and limitations were highlighted.

OpenSCAD® has an intuitive scripting language, very similar to C++, and powerful, as it allows to build even very complex 3D objects. In addition, the viewer next to the text editor allows the designer to immediately see the result produced by the coding and check for any errors, thanks to the preview.

An example of a perfect application for OpenSCAD® is the modeling of precise objects to be manufactured with a 3D printer.

OpenSCAD® is good for procedural/parametric modeling, but on the other hand, it does not give the possibility to go beyond the geometric model; it does not go beyond 3D; so it does not add semantic information to the objects produced, and it is not advisable for the application in BIM environment.

Therefore, eventually, it was needed to use Rhinoceros®. This turned out to be a great CAD modeling tool and thanks to the implementation of its Grasshopper® plugin, also a great tool for procedural/parametric modeling. Grasshopper® has the advantage of using elementary elements already coded to be used within a logical graph, which makes coding less tiring and even more intuitive. In addition, the possibility of using pieces of code in GHPython allows to create new logical elements to be inserted in the graph from scratch, making it another powerful tool in the hands of the designer.

Finally, the creation of objects with semantic information, ready to be used in a BIM environment, has been made possible through VisualARQ®, another Grasshopper® plugin. Indeed, VisualARQ® allows to associate IFC standards to Rhino6 objects, speaking of which, it should be noted that some problems have arisen.

First of all, IFC standards are well developed with regard to vertical structural elements, i.e., columns, pillars, walls, and architectural elements, i.e., windows, doors, railings, stairs, and roofs, but on the other hand, the development of standards referring to horizontal elements, such as those that constitute road infrastructure of all kinds, is still in progress.

And although the latter is provided for in the IFC 4 standard, it must also be said that several BIM protocols, such as Rhinoceros® and BIMcollab ZOOM®, refer only to the use of the IFC 2×3 exchange format.

To deal with this problem, each object of the railway infrastructure has been assigned the most appropriate IFC elements that correspond to its structural and functional characteristics, for example, to the cross beams correspond the beams, to the structural layers correspond the slab.

As mentioned, both Rhinoceros® and BIMcollab ZOOM® refer to the IFC 3×2 standard as the interchange format, yet a compatibility problem has emerged. In fact, even though in Rhinoceros® the materials have been assigned to each object created, by opening the file exported to IFC 3×2 with BIMcollab ZOOM®, this information is lost. The solution to this problem was to open the IFC 3×2 file with a text editor and modify its code adding information related to the fields of interest.

7. Conclusions

This study allowed highlighting the potential of a software system with the possibility to work in a BIM environment but still not mature enough for the design of railway infrastructure. At the current time, it is up to the designer to effectively choose and apply the most appropriate tools for the projects' purposes. Preferably and desirably, the regulatory provisions and previous personal knowledge should

be multidisciplinary and open to multiple coding languages. Much still needs to be performed in the direction of interoperability, the creation of libraries of objects and programs dedicated to the design of linear infrastructures.

Data Availability

The programming codes used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Kenley, T. Harfield, and A. Behnam, "BIM interoperability limitations: Australian and Malaysian rail projects," in *Proceedings of the 4th International Building Control Conference 2016 (IBCC 2016)*, vol. 7, p. 66, Kuala Lumpur, Malaysia, March 2016.

[2] G. Dell'Acqua, S. Guerra de Oliveira, and S. A. Biancardo, "Railway-BIM: analytical review, data standard and overall perspective," *Ingegneria Ferroviaria*, vol. 73, no. 11, pp. 901–923, 2018.

[3] McGraw-Hill Construction, *The Business Value of BIM for Infrastructure: Addressing America's Infrastructure Challenges with Collaboration and Technology*, Smart Market Report, New York, NY, USA, 2012.

[4] M. Borrmann, M. König, C. Kock, and J. Beetz, *Building Information Modeling–Technology Foundations and Industry Practice*, Springer, New York, NY, USA, 2018.

[5] M. Belsky, R. Sacks, and I. Brilakis, "Semantic enrichment for building information modeling," *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, no. 4, pp. 261–274, 2016.

[6] S. A. Biancardo, A. Capano, S. Guerra de Oliveira, and A. Tibaut, "Integration of BIM and procedural modeling tools for road design," *Infrastructures*, vol. 5, no. 37, 2020.

[7] S. A. Biancardo, N. Viscione, C. Oreto, R. Veropalumbo, and F. Abbondati, "BIM approach for modeling airports terminal expansion," *Infrastructures*, vol. 541 pages, 2020.

[8] F. Abbondati, S. A. Biancardo, S. Palazzo, F. S. Capaldo, and N. Viscione, "I-BIM for existing airport infrastructures," *Transportation Research Procedia*, vol. 45, pp. 596–603, 2020.

[9] S. A. Biancardo, F. Russo, R. Veropalumbo, V. Vorobjovas, and G. Dell'Acqua, "Modeling roman pavements using heritage-BIM: a case study in Pompeii," *The Baltic Journal of Road and Bridge Engineering*, vol. 15, no. 3, pp. 34–46, 2020.

[10] S. Fabozzi, S. A. Biancardo, R. Veropalumbo, and E. Bilotta, "I-BIM based approach for geotechnical and numerical modelling of a conventional tunnel excavation," *Tunnelling and Underground Space Technology*, vol. 108, Article ID 103723, 2021.

[11] A. S. Trbovich, A. Vučković, and B. Drašković, "Industry 4.0 as a lever for innovation: review of Serbia's potential and research opportunities," *Ekonomika Preduzeća*, vol. 68, no. 1-2, pp. 105–120, 2020.

[12] L. D'Acierno, M. Gallo, B. Montella, and A. Placido, "Analysis of the interaction between travel demand and rail capacity constraints," *WIT Transactions on the Built Environment*, vol. 128, pp. 197–207, 2012.

[13] M. Ercolani, A. Placido, L. D'Acierno, and B. Montella, "The use of microsimulation models for the planning and

management of metro systems," *WIT Transactions on the Built Environment*, vol. 135, pp. 509–521, 2014.

[14] L. D'Acierno, A. Placido, M. Botte, and B. Montella, "A methodological approach for managing rail disruptions with different perspectives," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 10, pp. 80–86, 2016.

[15] M. Botte, C. Di Salvo, C. Caropreso, B. Montella, and L. D'Acierno, "Defining economic and environmental feasibility thresholds in the case of rail signalling systems based on satellite technology," in *Proceedings of the 16th IEEE International Conference on Environment and Electrical Engineering 2016 (EEEIC 2016)*, pp. 1–5, IEEE, Florence, Italy, June 2016.

[16] L. D'Acierno, M. Botte, M. Gallo, and B. Montella, "Defining reserve times for metro systems: an analytical approach," *Journal of Advanced Transportation*, vol. 2018, pp. 1–15, Article ID 5983250, 2018.

[17] L. D'Acierno, M. Botte, and G. Pignatiello, "A simulation-based approach for estimating railway capacity," *International Journal of Transport Development and Integration*, vol. 3, no. 3, pp. 232–244, 2019.

[18] H. Ehrbar, "Building Information Modelling-a new tool for the successful implementation of major projects of German railways/building information modelling-ein neues werkzeug zur erfolgreichen realisierung von großprojekten der deutschen bahn," *Geomechanics and Tunnelling*, vol. 9, no. 6, pp. 659–673, 2016.

[19] A. G. Deutsche Bahn, "Anlage 3a zur Vereinbarung zur Finanzierung von Building Information Modeling (BIM) bei Pilotprojekten im Rahmen des Zukunftsinvestitionsprogramms," 2016.

[20] M. Bensalah, A. Elouadi, and H. Mharzi, *"Bim Integration to Railway Projects–Case Study"*, Railway Information Modeling RIM: The Track to Rail Modernization, New York, NY, USA, 2019.

[21] Trimble, "Sets the standard for BIM in railway projects," 2016, http://www.novapoint.com/sets-standard-bim-railway-projects.

[22] J. Neves, Z. Sampaio, and M. Vilela, "A case study of BIM implementation in rail track rehabilitation," *Infrastructures*, vol. 4, no. 1, 2019.

[23] R. Davies, F. Crespin-Mazet, and A. Linne, "BIM in Europe: innovation networks in the construction sectors of Sweden, France and the UK," in *Proceedings of 31st Annual ARCOM Conference*, vol. 1135, pp. 7–9, Lincoln, UK, September 2015.

[24] F. Abbondati, S. A. Biancardo, and G. Sicignano, "BIM parametric modelling of a railway underpass," *Ingegneria Ferroviaria*, vol. 75, no. 6, p. 443, Article ID 2016459, 2020.

[25] S. Smith, "Building information modelling - moving Crossrail, UK, forward," *Proceedings of the Institution of Civil Engineers-Management, Procurement and Law*, vol. 167, no. 3, pp. 141–151, 2014.

[26] I. May, M. Taylor, and D. Irwin, "Crossrail: a case study in BIM," 2017, https://ethz.ch/content/dam/ethz/special-interest/baug/igt/tunneling-dam/kolloquien/2016/May_Crossrail_BIM_Implementierung_ohne_Risiko.pdf.

[27] M. Ball, "On the fast track: high speed 2 uses BIM to connect cities across the UK," 2015, https://www.autodesk.com/redshift/high-speed-2-bim-connect-uk-cities/.

[28] United Kingdom Department for Transport, *"High Speed Two: From Concept to Reality–Moving Britain Ahead*, https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/629380/high-%

20speed-two-from-concept-to-reality.pdf, United Kingdom Department for Transport, London, UK, 2017, https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/629380/high-%20speed-two-from-concept-to-reality.pdf.

[29] M. Stacy and J. Birbeck, "BIM for the northern hub ECS1 programme," *The Structural Engineer*, vol. 91, no. 11, pp. 50–57, 2013.

[30] H. Cho, K. H. Lee, and S. H. Lee, "Introduction of construction management integrated system using BIM in the Honam High-speed railway lot No. 4-2," in *Proceedings of the 28th ISARC*, p. 1300, Seoul, Korea, 1305.

[31] La Lettre du BIMB, "DatBIM: bouygues Immobilier accélère dans le BIM," 2016, http://www.lalettredubim.com/bim/datbim-bouygues-immobilier-acclre-dans-le-bim/.

[32] G. Foeillet, "SNCF Réseau BIM: enjeux et Actions en cours," 2016, https://aqtr.com/system/files/file_manager/mardi_pm_foeillet.pdf.

[33] Rete Ferroviaria Italiana (RFI), "Norme tecniche per la progettazione dei tracciati ferroviari," 2006.

[34] Official Gazette of the Republic of Slovenia, "Pravilnik o zgornjem ustroju železniških prog (Rules for the upper structure of railway lines)," *ZVZeIP*, vol. 1, 2018.

[35] S. A. Biancardo, C. Oreto, N. Viscione, and F. Russo, "BIM approach for smart infrastructure design and maintenance operations," *Transportation Systems for Smart, Sustainable, Inclusive and Secure Cities*, vol. 131, 2020.