An IFC Representation for Process-Based Cost Modeling

Eduardo Luís Isatto

Universidade Federal do Rio Grande do Sul, Porto Alegre, RS 90035-190, Brazil
isatto@ufrgs.br

Abstract. One of the applications of BIM is for cost estimating in construction projects. However, the usual practices regarding the integration between cost and product models with BIM have been limited to the quantity takeoff of physical production measures from a digital model. The cost estimates of the building and its components usually happen outside the BIM model, through the same cost estimate method that has been traditionally used in the construction industry. Production costs are estimated based on previously known consumption rates of resources and theirs acquisition costs, which are commonly stored in external databases using “ad hoc” representation schemas, without any further information exchange with the BIM model. Nonetheless, the most recent version of the ISO 16739 standard (the Industry Foundation Classes, addendum 2, IFC4-add2) has included a number of new classes and properties aimed to extend BIM scope much beyond the product model. However, up to the moment, no specific schema using these new classes has been proposed in order to incorporate cost model information into the BIM model. This study was aimed to investigate the possibility of representing process-based cost information using IFC 4 objects. The research method approach used was case study. Some use cases of processes cost models were selected from real cost modeling cases, and generalize to design a process-based cost modeling schema in IFC. The resulting schema was then instantiated in a prototype, used to validate the logic and output of the representation against manual calculations. The representation of process-based cost modeling proved to be capable of successfully merging the cost modeling schema within a BIM model. It opens the possibility of close integration between product and cost models with BIM technology, as well as the storage and dissemination of cost models and their related information using open and interoperable representation.

Keywords: Building Information Modeling (BIM) · IFC · Cost modeling · Data representation · Cost estimating

1 Introduction

Effective cost management is a key factor for the success of a construction project [14]. It involves much more than just estimating the monetary effort, requiring an active posture of the managers in planning and influencing the behavior of project costs to achieve the desired results.

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2021
E. Toledo Santos and S. Scheer (Eds.): ICCCBE 2020, LNCE 98, pp. 519–528, 2021.
https://doi.org/10.1007/978-3-030-51295-8_37

Cost estimation of construction projects faces many difficulties particularly due to the unique nature of the construction product, among other factors. However, along the last decade the BIM technology has proven to be a valuable approach to perform quantity take-offs (QTO) and cost estimates [10].

Due to the possibility of automatically calculate and assess production quantities based on the geometric characteristics of building elements, BIM software is particularly capable of tracking the effects in costs due to changes in design, thus providing an extreme valuable support to target-costing strategies as those implied in Integrated Project Delivery (IPD) [1].

However, the use of BIM technology for cost estimation rarely goes beyond the QTO procedure. Cost estimates are also dependent on production costs per unit (unit costs), which are usually calculated based on previously known consumption rates of resources and their acquisition costs. These are commonly stored in external databases using “ad hoc” representation schema, without any further information exchange with the BIM model. As a consequence, the cost estimating process results highly dependent on the particular software being used, and the exchange of resulting data is made extremely limited.

Since the most recent version of the ISO 16739 standard, the Industry Foundation Classes schema (the IFC4) includes several new classes and properties aimed to extend BIM scope much beyond the product model, particularly in terms of resources and task types. Nonetheless, except for some references in the buildingSMART documentation of IFC4-add2 [2], publications reporting the use of these new classes for cost management are still rare, and approaches to incorporate cost model information based on resource usage into the BIM model are still to be proposed.

There are many advantages in representing cost information as a part of the BIM model. Firstly, it extends the scope of the BIM model from Product Data to Project Management Data, as suggested by Froese [7]. As a consequence, project data can be stored in a single place, or segmented in many places without losing its integrity. Secondly, it can help managers to deal with trade-offs related to the project “iron triangle”—cost, scope, and time—by providing a transparent and solid integration among cost objects, product elements, and project tasks. Thirdly, adopting ISO 16739 standard (the Industry Foundation Classes) can assure strong compatibility between different software and across building life-cycle stages. This facilitates, for example, the exchange of project information among distinct applications dedicated to specialized tasks, as project-scheduling and cost estimating, or even between these applications and ERP systems. Fourthly, an IFC-based cost model can enable cost information to be consolidated and exchanged with other project members through a federated model. Finally, process-based cost information can be stored as part of process libraries to be used in the context of a specific project, company, or even as public processes libraries (e.g., the Brazilian National System of Construction Costs and Indexes Survey —SINAPI) [3].

Considering the relevance of the research problem, this study aims to contribute to reducing the knowledge gap by proposing a process-based representation for cost information using IFC 4 objects. With such representation, it is expected to further incorporate resource-usage cost information directly into IFC projects files, or as member-of an IFC federated model.

2 Background

2.1 Process-Based Cost Modeling

Cost estimates methods vary depending on information availability and the intended purpose. They can range from approximate values early in the design to more precise values after the design is complete [6]. During the early design phase, *conceptual estimating methods* are used, as rough-order-of-magnitude (ROM), based on cost per unity or the production capacity, or even associated with the area or volume of the building (e.g., cost per square foot). As the design matures, *detailed estimating methods* can be used to develop cost estimates, based on cost per component (unit costs of assemblies) or per unit of work result (unit cost of processes). While conceptual estimating methods are based on historical cost patterns, detailed methods usually rely on the causes of the costs, the resource usage patterns.

All these methods are acceptable if the intended purpose is product pricing, but only the latter can provide adequate support for cost management during project execution. The reason is simple: product-based cost models are structured to represent building components or finished building and are thus concerned more on ends than on means [12]. Conversely, process-based models are naturally coupled with the project progress as they are structured to represent tasks as the cost objects.

As process-based costing method relies on process patterns, they are much more flexible to model production costs in contexts where product patterns do not exist or are difficult to identify, as in Engineering-To-Order (ETO) environments like construction. For example, all cut-and-bent reinforced bar configurations employ up to three processes: cut, bend and tie.

For the sake of clarity, in this article is assumed that tasks are the means to achieve a desired product as a result, and that these tasks demand the use of resources like material, labor, equipment, and other products (Fig. 1). Therefore, costs of the resources are allocated primarily to the processes, and then to tasks and products, in that order.

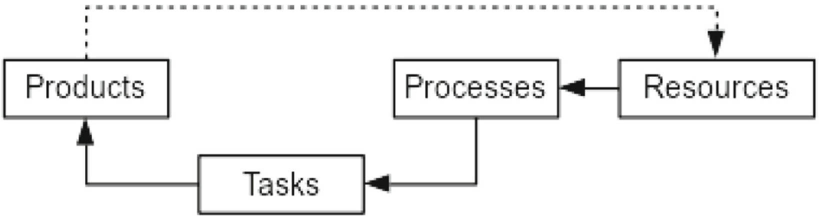
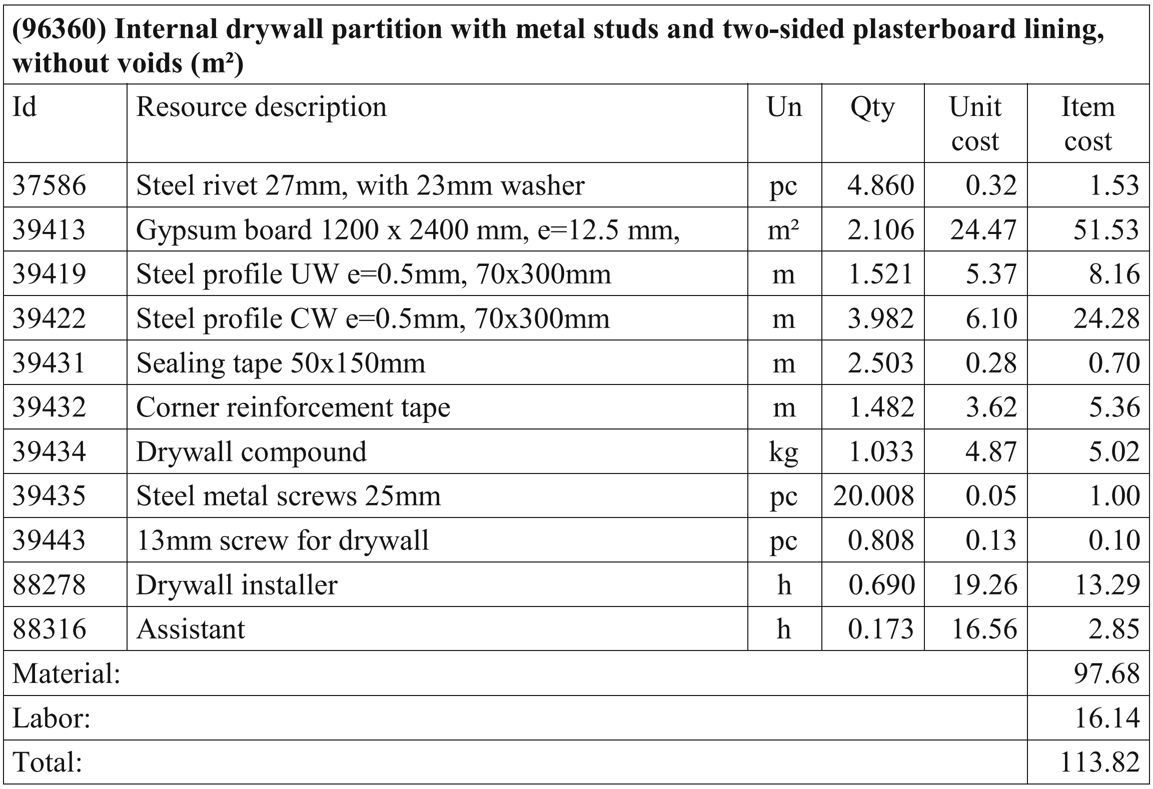


Fig. 1. Conceptual model for process-based costing

Process-based cost compositions are commonly used in construction industry. The Brazilian National System of Construction Costs and Indexes Survey (SINAPI) [3] uses these cost compositions to assess the unit price of all the public construction project, and its use is mandatory to all contractors. An example adapted from those cost compositions is presented in Fig. 2.



(96360) Internal drywall partition with metal studs and two-sided plasterboard lining, without voids (m²)

Id	Resource description	Un	Qty	Unit cost	Item cost
37586	Steel rivet 27mm, with 23mm washer	pc	4.860	0.32	1.53
39413	Gypsum board 1200 x 2400 mm, e=12.5 mm,	m²	2.106	24.47	51.53
39419	Steel profile UW e=0.5mm, 70x300mm	m	1.521	5.37	8.16
39422	Steel profile CW e=0.5mm, 70x300mm	m	3.982	6.10	24.28
39431	Sealing tape 50x150mm	m	2.503	0.28	0.70
39432	Corner reinforcement tape	m	1.482	3.62	5.36
39434	Drywall compound	kg	1.033	4.87	5.02
39435	Steel metal screws 25mm	pc	20.008	0.05	1.00
39443	13mm screw for drywall	pc	0.808	0.13	0.10
88278	Drywall installer	h	0.690	19.26	13.29
88316	Assistant	h	0.173	16.56	2.85
Material:					97.68
Labor:					16.14
Total:					113.82

Fig. 2. Example of cost composition, adapted from Brazilian National System of Construction Costs and Indexes Survey (SINAPI)

The example in Fig. 2 illustrates the cost estimate per square meter for a process of wall partition installation that includes both the steel framing assembly and plasterboard installation in one single process. This cost composition may be part of the project cost database as it may be used to estimate the cost to any drywall of this type that is installed in a similar context to the one used to assess quantities and costs of the resources.

2.2 IFC Representation

Most of the research on BIM-based cost estimating approaches have been based on the product model, particularly the quantity take-off (QTO) [5, 10, 11, 15]. Some researches as Lee et al. [9] and Niknam and Karshenas [13] suggested an ontology-based approach for cost modeling that included the representation of product and process models. Monteiro and Martins [11] alert that the quality of BIM-based quantity take-offs are very dependent on the authoring, and requires rules to be followed in order to guarantee the consistency of the quantities obtained. Using IFC could avoid software dependency for QTO but there are very few examples of IFC-based cost estimating alternatives. One of the first studies on the topic was from Staub-French et al. [15], who proposed a cost estimating system that directly uses IFC files and external prices databases to accomplish the cost estimation. More recently, Lawrence et al. [8] proposed the use of flexible mappings between BIM models in IFC and cost information. Nonetheless, all of these studies were purely based on the product model. Among them, the only exception is Lee et al. [9] and Niknam and Karshenas [13], which partially included process information but still reside in a higher level of abstraction as a result of the ontology-based approach chosen for the representation.

The problem approached in this research is how to represent these resources costs and allocate them to the processes entities using the IFC4 schema. The IfcTask class is part of the IFC schema since version 1.0, with the main purpose of combining BIM model and project schedules (also know as BIM 4D modeling) [2, 6]. Nonetheless, examples of IfcTask usage for scheduling as the example E.13 provided by buildingSMART [2] are very few, and most of the existing BIM 4D modeling applications seem to not take advantage of this IFC class, favoring proprietary solutions to combine the BIM product model and project schedules to create 4D models.

More recently, project management capabilities of the IFC were greatly extended in the version 4, with several new classes in the process extension schema, as well as in the shared and specific construction management domains. It is noteworthy the class IfcTaskType, included as a way to generalize task definitions without the concern with task dependencies, duration or schedule. BuildingSMART [2] describes IfcTaskType as follows:

> *"A IfcTaskType is a reference definition for a unit of work that may be broken down into (a sequence of) subtasks. [...] Usage of IfcTaskType defines the parameters for one or more occurrences of IfcTask. [...] Fig. 145 shows the definition of a task type that is part of a task template library."*

The buildingSMART [2] also gives some indication on possible strategies to create a task template library and to combine task types with construction resources types. Those strategies, however, are all limited to very simple examples or hypothetical situations, and do not provide a comprehensive and complete approach on how to represent process-based cost compositions in the IFC file.

3 Research Approach

The first step taken was to design a preliminary approach to represent process-based cost compositions using the more recent IFC schema (IFC-add2). This was based on literature review and the strategies suggested by buildingSMART [2] for task template libraries and construction resource types entities.

To test the representation, the schema was instantiated in a prototype, used to extend the original IFC file in the EXPRESS notation with tasks, resources, and cost information. This file was then used to calculate the cost of the task and the associated wall, based on resources unit cost and usage, as well as the size of the wall.

The process-based cost composition data used for the test was the one shown in Fig. 2, associated with the installation of drywall internal partitions. This particular composition was chosen due the number and diversity of its resources, and also because it is a widely adopted and well-documented process in many countries.

That specific cost composition from the Brazilian National System of Construction Costs and Indexes Survey (SINAPI) is part of a public database and freely available without any cost.

The BIM model used for the test was chosen among well-document examples on the use of the software ifcopenshell and IFC4 schema. The chosen model is the resulting output of an example created by Kianwee Chen and Patrick Janssen [4] to demonstrate how to generate IFC files using ifcopenshell and python. The example is named "Creating a simple wall with property set and quantity information", and freely available at the IfcOpenShell Academy (http://academy.ifcopenshell.org/).

The test used ifcopenshell-python 0.60 and python 3.6.9 for creating the initial file, to include the necessary tasks, resources and cost information, and to perform calculation. The example was slightly modified for the sake of compatibility of the resulting file ('hello_wall.ifc') with the chosen software.

The test consisted in two steps. In the first, the original IFC file 'hello_wall.ifc' was read using a python script and ifcopenshell, and then the task, resources and cost information from the cost composition were inserted into the IFC model, which was them written as a new file 'wall_with_cost.ifc', in the EXPRESS notation.

Ifcopenshell was used in this step also to assure a minimum validity of the modified file against the IFC 4x2 schema. A python script was developed to read and modify the IFC EXPRESS file.

The second steps consisted in reading the file 'wall_with_cost.ifc' and performing the cost calculation of the unit cost of the process, and the cost of the task and the resulting product (the wall). In this test, only one task was associated with the wall.

4 Results

Figure 3 presents the strategy chosen in this research to represent process-based costs in the IFC schema. Basically, costs are allocated to tasks through the unit cost of the resources, which are represented as generic resources (resource types). These entities do not represent the resources themselves but their use for some purpose. Thus, both the unit costs and the base quantities can be represented as attributes of the resource entity. The unit costs can be associated with existing entities in the IFC project or library that play the roles of these resources, or may originate from external references as databases directly to the resources without the intervention of an existing entity. In this study, all the information about resources costs and quantities to be used in the chosen task were hard-coded in the python script, assuming that they do not result from any other entity of the model.

The processes were modeled as task types, as processes can be seen as generic tasks. The resource types and their attributes may be associated with the processes using the objectified relationship 'IfcRelAssignsToProcess'. With these entities, all the needed elements to calculate the cost of the composition are then defined.

It is noteworthy that processes (task types) may need to be associated with generic products (product types) when these products are needed as resources for other processes. For example, the process of preparing mortar results in a generic product (mortar), that may be further used as a resource for other processes (e.g. to execute brickwork).

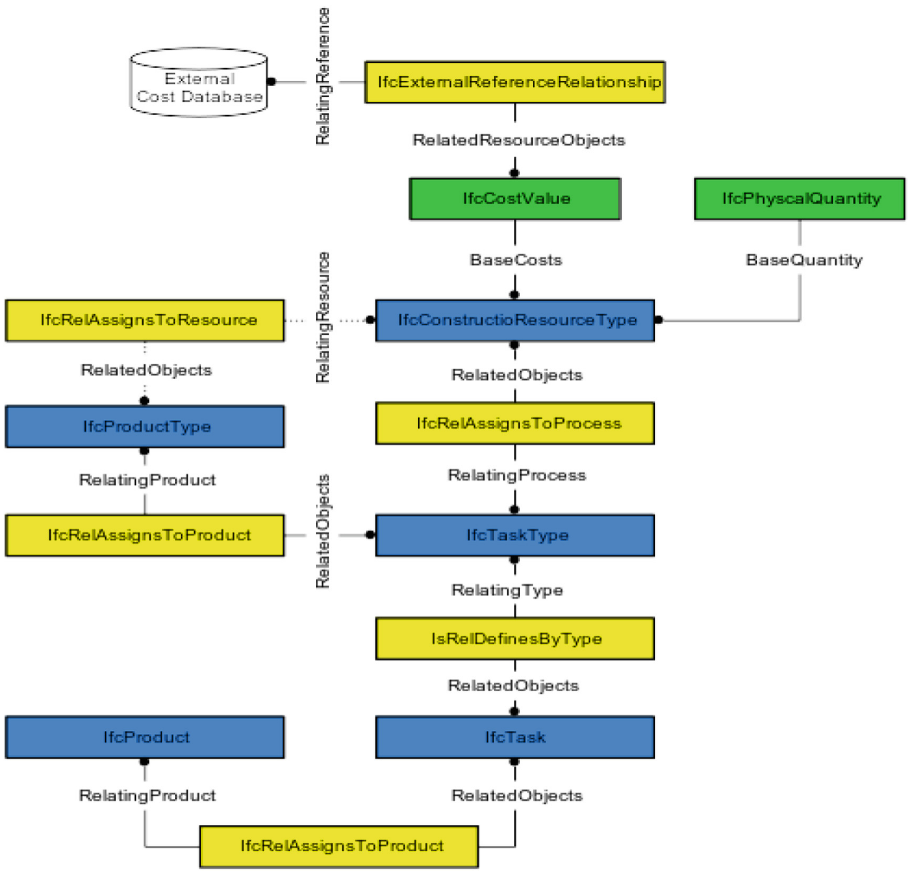


Fig. 3. Proposed IFC representation for process-based cost modeling

In a further step, the cost of the associated tasks and products can be calculated by multiplying the composition unit cost by the number of physical quantities in the associated product. In this study, the relevant attribute of the wall is its area, which is multiplied by the cost per square-meter of the composition, resulting cost of the task 'Installing the wall'. This task cost is then added to the cost of the wall. In this study only one task was applied to the product. However, it is very important to differentiate the cost of the task from the product cost as a single product can be associated with more than one task (e.g. to clean the surface, to sand, and to paint). This is particularly relevant when tasks do not create (or destroy) product entities.

The proposed schema does not include the instantiation of the IfcProductType in the IfcProduct. That instantiation is possible and may simplify significantly the effort in associating tasks and products instances, but would require that all products were represented by a resource type, which may not be feasible in many circumstances.

The first step was to include task, resource and cost data into the 'hello_wall.ifc' file. A python script was created to read the file, to create IFC entities representing tasks, resources, costs and the relationship entities shown in Fig. 3, using the data presented in Fig. 2. The file was then written to disk with a new name ('wall_with_-cost.ifc'). An excerpt of that file presenting the additional entities is shown in Fig. 4.

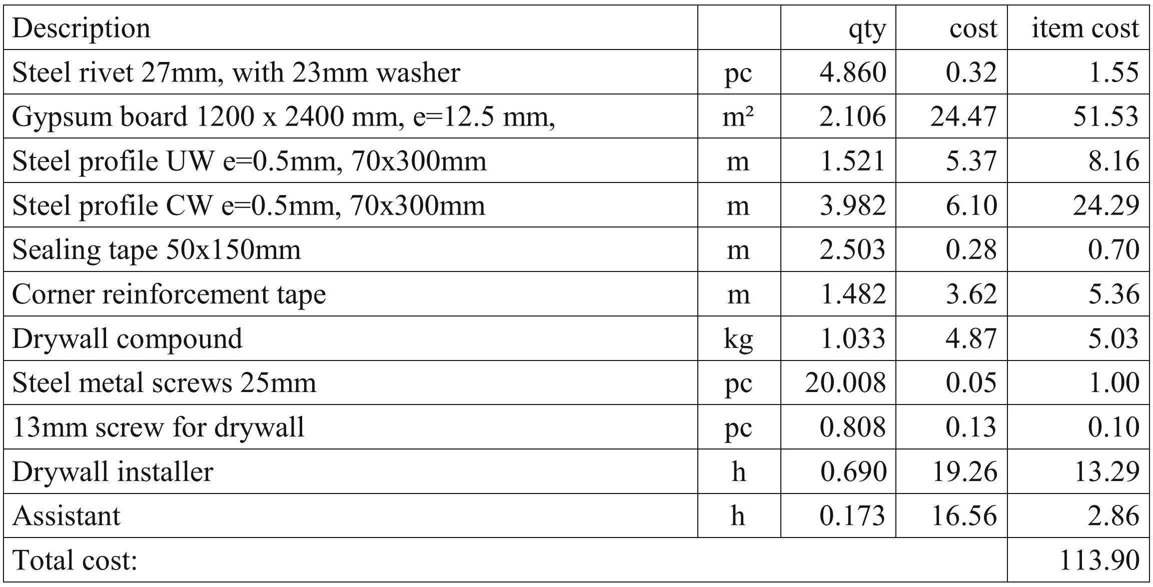
The second step consisted in reading the new IFC file, extracting task, resource and cost information, to calculate the unit cost of the process (task type). The cost of the task 'Install partition wall' was then derived from the unit cost of the process used

```
#128=IFCMONETARYUNIT('BRL');
#129=IFCTASKTYPE('0C5V8g_aD7hvY6ZAvRTda0',#5,'96360','Internal  drywall  partition  with  metal
      studs and two-sided plasterboard lining, without voids (m2)',$,$,$,$,$,.CONSTRUCTION.,$);
#130=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(0.32),$,$,$,$,$,$,$);
#131=IFCQUANTITYCOUNT('*',$,$,4.86,$);
#132=IFCCONSTRUCTIONMATERIALRESOURCETYPE('0m9w5nnA961vcFDQmAMtQO',#5,'37586','Steel  rivet  27mm,
      with 23mm washer (pc)',$,$,$,$,$,(#130),#131,.NOTDEFINED.);
#133=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(24.47),$,$,$,$,$,$,$);
#134=IFCQUANTITYCOUNT('*',$,$,2.106,$);
#135=IFCCONSTRUCTIONMATERIALRESOURCETYPE('0wUJ9ZeAL6TBIFfmN0cStK',#5,'39413','Gypsum  board  1200
      x 2400 mm, e=12.5 mm (m2)',$,$,$,$,$,(#133),#134,.NOTDEFINED.);
#136=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(5.37),$,$,$,$,$,$,$);
#137=IFCQUANTITYCOUNT('*',$,$,1.521,$);
#138=IFCCONSTRUCTIONMATERIALRESOURCETYPE('3ZWv3peU93TR5f1iJrdRP2',#5,'39419','Steel  profile  UW
      e=0.5mm, 70x300mm (m)',$,$,$,$,$,(#136),#137,.NOTDEFINED.);
#139=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(6.1),$,$,$,$,$,$,$);
#140=IFCQUANTITYCOUNT('*',$,$,3.982,$);
#141=IFCCONSTRUCTIONMATERIALRESOURCETYPE('0LhMs3lI18hB6zIYQFKNQ_',#5,'39422','Steel  profile  CW
      e=0.5mm, 70x300mm (m)',$,$,$,$,$,(#139),#140,.NOTDEFINED.);
#142=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(0.28),$,$,$,$,$,$,$);
#143=IFCQUANTITYCOUNT('*',$,$,2.503,$);
#144=IFCCONSTRUCTIONMATERIALRESOURCETYPE('2A47nsYcj6GBNfFZ4nq3UJ',#5,'39431','Sealing      tape
      50x150mm (m)',$,$,$,$,$,(#142),#143,.NOTDEFINED.);
#145=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(3.62),$,$,$,$,$,$,$);
#146=IFCQUANTITYCOUNT('*',$,$,1.482,$);
#147=IFCCONSTRUCTIONMATERIALRESOURCETYPE('29yTWkxcn9aRIBIaDhIyAs',#5,'39432','Corner  reinforce-
      ment tape (m)',$,$,$,$,$,(#145),#146,.NOTDEFINED.);
#148=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(4.87),$,$,$,$,$,$,$);
#149=IFCQUANTITYCOUNT('*',$,$,1.033,$);
#150=IFCCONSTRUCTIONMATERIALRESOURCETYPE('08LRSU__bBhvciUfFGRhZQ',#5,'39434','Drywall   compount
      (kg)',$,$,$,$,$,(#148),#149,.NOTDEFINED.);
#151=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(0.05),$,$,$,$,$,$,$);
#152=IFCQUANTITYCOUNT('*',$,$,20.008,$);
#153=IFCCONSTRUCTIONMATERIALRESOURCETYPE('0wB$xLMcj2Fv7n9922LpHe',#5,'39435','Steel metal screws
      25mm (pc)',$,$,$,$,$,(#151),#152,.NOTDEFINED.);
#154=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(0.13),$,$,$,$,$,$,$);
#155=IFCQUANTITYCOUNT('*',$,$,0.808,$);
#156=IFCCONSTRUCTIONMATERIALRESOURCETYPE('0V$h_yJuH45f_X8hGIk0ss',#5,'39443','13mm   screw   for
      drywall (pc)',$,$,$,$,$,(#154),#155,.NOTDEFINED.);
#157=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(19.26),$,$,$,$,$,$,$);
#158=IFCQUANTITYCOUNT('*',$,$,0.69,$);
#159=IFCCONSTRUCTIONMATERIALRESOURCETYPE('1$rUFnYlbFFfxjoiQBtEzJ',#5,'88278','Drywall  installer
      (h)',$,$,$,$,$,(#157),#158,.NOTDEFINED.);
#160=IFCCOSTVALUE($,$,IFCMONETARYMEASURE(16.56),$,$,$,$,$,$,$);
#161=IFCQUANTITYCOUNT('*',$,$,0.173,$);
#162=IFCCONSTRUCTIONMATERIALRESOURCETYPE('0xtuEhyo98QuVIxiXGy$8N',#5,'88316','Assistant
      (h)',$,$,$,$,$,(#160),#161,.NOTDEFINED.);
#163=IFCRELASSIGNSTOPROCESS('0eW3obryH8HRSJfkvMqB9y',#5,$,$,(#132,#135,#138,#141,#144,#147,#150,
      #153,#156,#159,#162),$,#129,$);
#164=IFCTASK('27l7nTkmrAd8WEPkypnkFP',#5,'0','Construction',$,$,$,$,$,$,$,$,$);
#165=IFCTASK('0iRNnNJyz448v9vZoLXTGg',#5,'1','Install partition wall',$,$,$,$,$,$,$,$,$);
#166=IFCRELDEFINESBYTYPE('1Kq088T6jAwOJB7JmASwZC',#5,$,$,(#165),#129);
#167=IFCRELNESTS('1BnZkzUEvDuOekA_S65OlB',#5,$,$,#164,(#165));
#168=IFCRELASSIGNSTOPRODUCT('0EJmNphm93qu5B5JzVYv1Z',#5,$,$,(#165),$,#66);
```

Fig. 4. An excerpt of the new IFC EXPRESS file with the new additional entities created to represent tasks, resources and cost information

(entity #129, 'Internal drywall partition with metal studs and two-sided plasterboard lining, without voids (m2)') and the area of the associated wall (entity #66, 'An awsome wall'). The resulting information is presented in Fig. 5.

A comparison between Figs. 2 and 5 shows that most of the values as identical, demonstrating that the cost composition information was effectively stored and retrieved using the IFC schema and EXPRESS notation.



Description		qty	cost	item cost
Steel rivet 27mm, with 23mm washer	pc	4.860	0.32	1.55
Gypsum board 1200 x 2400 mm, e=12.5 mm,	m²	2.106	24.47	51.53
Steel profile UW e=0.5mm, 70x300mm	m	1.521	5.37	8.16
Steel profile CW e=0.5mm, 70x300mm	m	3.982	6.10	24.29
Sealing tape 50x150mm	m	2.503	0.28	0.70
Corner reinforcement tape	m	1.482	3.62	5.36
Drywall compound	kg	1.033	4.87	5.03
Steel metal screws 25mm	pc	20.008	0.05	1.00
13mm screw for drywall	pc	0.808	0.13	0.10
Drywall installer	h	0.690	19.26	13.29
Assistant	h	0.173	16.56	2.86
Total cost:				113.90

Fig. 5. Calculation results for the cost of the task 'Install partition wall' based on information retrieved from the IFC file.

5 Conclusion

The article presents an initial study to represent process-based cost models using IFC4x2 schema and EXPRESS notation. Based on literature review and examples from buildingSMART, an IFC schema for representing process-based cost models was proposed. The proposed schema was instantiated in an IFC file based on a publicly available BIM model developed for demonstration purposes, which was populated with real cost data from the Brazilian National System of Construction Costs and Indexes Survey (SINAPI). The resulting IFC file was used to retrieve cost model information, and to calculate the cost of process, task and product instances. The results confirmed the effectiveness of the proposed representation to be capable of successfully integrating the proposed cost modeling schema within a BIM model. It opens the possibility of close integration between product and cost models with BIM technology, as well as the storage and dissemination of cost models and their related information using open and interoperable representation.

References

1. Ballard, G., Reiser, P.: The St. Olaf college fieldhouse project: a case study in designing to target cost. In: Bertelsen, S., Formoso, C. (eds.) 12th Annual Conference of the International Group for Lean Construction. Elsineor, Denmark (2004)
2. buildingSMART: Industry Foundation Classes -Version 4.2 bSI Draft Standard (2019). https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML. Accessed 06 Feb 2020
3. CAIXA (Caixa Econômica Federal): Sistema Nacional de Pesquisa de Custos e Índices da Construção (SINAPI), Relatório de Insumos e Composições – Dez/19 - Sem Desoneração

(2019). http://www.caixa.gov.br/site/paginas/downloads.aspx#categoria_556. Accessed 06 Feb 2019

4. Chen, K., Janssen, P.: Creating a simple wall with property set and quantity information (2015). http://academy.ifcopenshell.org/creating-a-simple-wall-with-property-set-and-quantity-information. Accessed 06 Feb 2020
5. Choi, J., Kim, H., Kim, I.: Open BIM-Based quantity take-off system for schematic estimation of building frame in early design stage. J. Comput. Des. Eng. **2**(1), 16–25 (2015)
6. Eastman, C., Teicholz, P., Sacks, R., Liston, K.: BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, 2nd edn. John Wiley, Hoboken (2011)
7. Froese, T.: Future directions for model-based interoperability. In: Construction Research Congress, p. 1–8. American Society of Civil Engineers, Reston (2003)
8. Lawrence, M., Pottinger, R., Staub-French, S., Nepal, M.: Creating flexible mappings between building information models and cost information. Autom. Constr. **45**, 107–118 (2014)
9. Lee, S., Kim, K., Yu, J.: BIM and ontology-based approach for building cost estimation. Autom. Constr. **41**, 96–105 (2014)
10. Mattern, H., Scheffer, M., König, M.: BIM-Based quantity take-off. In: Borrmann, A., König, M., Koch, C., Beetz, J. (eds.) Building Information Modeling. Springer International Publishing (2018)
11. Monteiro, A., Martins, J.: A survey on modeling guidelines for quantity takeoff-oriented BIM-Based design. Autom. Constr. **35**, 238–253 (2013)
12. Nguyen, H., Tommelein, I., Ballard, G.: Process-Based Cost Modeling to Support Lean Project Delivery. In: Tzortzopoulos, P., Kagioglou, M. (eds.): 16th Annual Conference of the International Group for Lean Construction, p. 577–588. Manchester, UK. (2008)
13. Niknam, M., Karshenas, S.: Integrating distributed sources of information for construction cost estimating using semantic web and semantic web service technologies. Autom. Constr. **57**, 222–238 (2015)
14. Pollack, J., Helm, J., Adler, D.: What is the iron triangle, and how has it changed? Int. J. Managing Projects Bus. **11**(2), 527–547 (2018)
15. Staub-French, S., Fischer, M., Kunz, J., Paulson, B.: A generic feature-driven activity-based cost estimation process. Adv. Eng. Inform. **17**(1), 23–39 (2003)