

# The Devil in the Details: Fine-Grained Enterprise Model Weaving

David Naranjo<sup>(✉)</sup>, Mario Sánchez, and Jorge Villalobos

Department of Systems and Computing Engineering,  
Universidad de Los Andes, Bogotá, Colombia  
{da-naran,mar-san1,jvillalo}@uniandes.edu.co

**Abstract.** When developing Enterprise Models, it is common to aggregate information from several partial models that describe a fragment of the enterprise. This integration is made by connecting elements from different domain models, and is usually a manual task for two reasons: (1) the criteria for connecting pairs of elements is mostly subjective and requires specialized domain knowledge, and (2) any error may impact the coherency of the whole model. However, manual weaving is a difficult and tedious task, with limited support both from methodologies and tools alike. In this paper, we describe a visual approach for weaving domain models, connecting model elements by using adjacency matrices, and visualizing changes at the moment they are made.

**Keywords:** Enterprise Architecture · Enterprise Models · Business Models · Composition

## 1 Introduction

Enterprise Models are the cornerstone of model-based Enterprise Architecture and its related fields. These models cover different dimensions that come across an organization, such as the Business, Application and Technology dimensions. In turn, each dimension can be subdivided into several knowledge domains, at varying levels of abstraction: For instance, we can identify domains such as Strategy, Processes, Value Stream, Products, Organization Structure, among several others, as part of the Business Dimension.

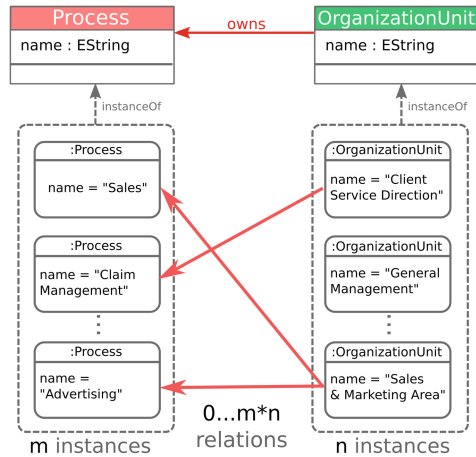
However, with more widespread use and more powerful modeling tools available, we now have two contradictory needs: On one hand, we need small, manageable artifacts to **model** the organization with a reduced set of concerns at the time. On the other hand, in order to answer complex questions through their **analysis**, we need really big and detailed models that give the sense of wholeness to the organization, and enable the examination of cross-cutting properties, such as alignment and change impact.

Under this light, we can identify two tendencies: (1) Heavyweight, established EA Frameworks such as Zachman, TOGAF, and EBMM, that propose a fixed (though in some cases extensible with profiles) metamodel, and (2) More

recent, customizable approaches, such as Multi-perspective Enterprise Modeling [7], which aim to incrementally construct the Enterprise Model with multiple extensible domain-specific metamodels.

The former are the product of a group of experts that suggest a modeling method, as well as relevant concepts and their relationships, and aim for a general, one-size-fits-all description of the enterprise. Heavyweight metamodels manage the complexity of modeling by using multiple Views of the model (which are models by their own rights), and offer a limited set of analysis techniques. On the other hand, Multi-perspective approaches argue that there is no easy way to describe every organization by using the same criteria, given their different business needs and maturity. Their flexibility translates into more freedom for designing and analyzing their respective models, but their development can be more expensive, and require more experience and knowledge to accurately describe an organization.

Independent of the approach, we can suggest that Enterprise Models are just a snapshot of the state of an organization in a given instant of time, so it is always subject to change, and it can happen at the instance level or at the meta-level. For example, we can add a new Process at the instance (model) level, or we can enhance the definition of what a Process is, –e.g. to now include an ownership relationship to an OrganizationalArea (see Fig. 1)– at the metamodel level.



**Fig. 1.** A single mapping of two concepts at the meta-level can result in  $m \cdot n$  relationships at the instance level.

Instance-level modifications are trivial, as it just means that we create, delete, or update an instance or relation of the model. Meta-level change is more troublesome, as we need to consider that changes in the metamodel affect the models that are produced with it. Again, this issue has been approached by several authors, and nowadays we have several languages and tools to perform automated (or semi-automated) weaving of models. However, we have to consider

that these approaches are not totally accurate, as they rely on techniques –such as textual comparison of instance attributes– that can miss several matches, for instance, the link between the *Sales and Marketing Area* and the *Advertising Process* in Fig. 1, or make incorrect mappings, like adding a relationship from the *General Management Unit* and the *Claim Management Process*.

Thus, the responsibility of discovering and correcting mistakes in the weaving process falls entirely in the modeler and his ability to grasp the whole model inside his head. This can be a complex and daunting task, even for the smallest of the cases: relating two concepts from different domains at the meta-level can result in a considerable number of relationships (see Fig. 1). This problem is more evident in situations when we must be careful with the details, e.g. when composing multiple domain models that describe different (but related) aspects of the enterprise, without affecting the semantics of individual metamodels.

In this paper, we will argue that current modeling and weaving tools do not work so well under these circumstances, as they do not offer enough information for the modeler to assess if the weaving is being done correctly at any stage of the process. In order to give immediate feedback when modeling, we consider that Visual Analysis techniques can provide a better support for this task, as well as reduce the effort of the modeler. With this in mind, we propose the use of a set of coordinated views to visualize this weaving process and the elements involved, complemented with the use of an adjacency matrix that facilitates the insertion and removal of relationships at the instance level.

In order to explain our approach, Sect. 2 will deepen on the characterization of Fine-grained model weaving, and introduce a concrete use case, the integration of Business Domains. Section 3 will describe similar approaches, both with general purpose modeling tools, as well as specialized weaving approaches. Then, Sect. 4 will describe our proposed approach, as well as the tool that implements it. Finally, Sect. 5 will illustrate the use of the tool for combining five Business Domains of a Social Networking service.

## 2 Fine-Grained Weaving of Multiple Domains: The Case of Business Models

In order to design, implement, and manage software and technological artifacts, as well as to integrate them to the existing organization structure and infrastructure, we make use of several domain methods that encapsulate domain knowledge. For instance, it is common for us to use Business Process Model and Notation (BPMN) diagrams to describe the Roles, Processes, Tasks, and control flows of a given Core Process, or ArchiMate models to describe the Business, Information, and Technological domains from a high level of abstraction, as well as how they are interconnected.

Each of these models are, of course, limited in scope, and usually describe -in varying levels of detail- a small dimension of a real-world system. In fact, *we want* these models and their metamodels to remain small, as it allows us to break the enterprise into manageable parts. After all, modeling is a human activity, and

we have problems visualizing –grasping– hundreds of components of the model in the same view.

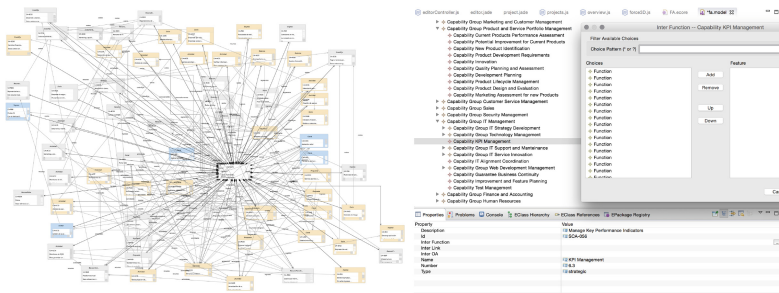
Not only we make use of multiple languages; we create new ones, that can be based on existing languages that are too complex, or even brand-new languages for domains that haven’t been formalized yet. The development of these tailored metamodels has been gaining traction, as it allows flexibility, modularity, and reuse, and contrasts with one-size-fits-all perspective of early business and enterprise metamodels.

### 2.1 Business Models

Recently, there has been a lot of interest around the integration of Business Models (BMs) and Enterprise Models [8, 14], in order to add coherency to the business dimension of the latter. Ultimately, BMs are a description of the value creation dynamics of an enterprise, describing phenomena that cannot be explained with just its operation, thus becoming a valuable knowledge asset.

Literature on Business Models [16] usually starts by acknowledging that we cannot express how an organization creates value from an individual viewpoint, and several authors have attempted to formalize the term by describing the different components of a Business Model.

For instance, Osterwalder et al. [13] propose a group of nine subject areas that are critical for the proposition of a Business Model, and are the building blocks of the Business Model Canvas –BMC– [12]. They also underscore the need to formalize concepts and provide a common language through *“Rigorously defined meta-models of business models in the form of formal reference models or ontologies”*.



**Fig. 2.** General purpose views for editing models. **Left:** A visual editor showing a BMC model. **Right:** A tree editor of a Business Model.

However, existing Business Model ontologies, such as the BMC, are mainly descriptive in nature, which means they can be valuable brainstorming tools, but they are not helpful when it comes to implementing BMs [15], as well as to promote experimentation and innovation with those models [2], e.g. by extending

their metamodels with other domains. Moreover, constructs such as Strategy, Ecosystem, Business Processes, and Partnership Network are outside of the scope of the BMC, but have a strong relation to Business Models.

With the purpose of exemplifying the need of fine-grained model weaving, we will consider the case of Business Models, which will be our concern for the rest of this paper.

## 2.2 *Fine-grained Weaving*

We can assert that BMs embed several **domains** (or perspectives) that are of our interest in the context of Enterprise Modeling. Moreover, depending on their context, some aspects can be given more or less detail; for instance, a manufacture company may require Product and Provisioning Models to describe its intent, while a banking company may require detailed information of its financial model, to provide an accurate description of its business.

Our approach for integrating and analyzing these separate domains consists in providing several visual aids that lower the cognitive burden of the addition and deletion of inter-domain relationships, as well as offering a way for diagnosing the quality of this weaving. Please note that this is a particular method that makes sense only when we have multiple domain models that we wish to integrate into an Enterprise Model. We associate elements from different domains because, from the perspective of the modeler, they are related in some way, even when there is no explicit connection by comparing their attributes (see Fig. 1).

In summary, we will use the term *Fine-grained Model Weaving* to refer to the lightweight **integration of different models** under these conditions:

1. This is a sensible task that requires an expert, as errors or missing relationships affect the quality of the integrated model.
2. The weaving at the instance level is considerably complex (i.e. the possible number of inter-domain relations is large).
3. It is mostly manual, but can be assisted by automatic and semi-automatic weaving tools.
4. The modeler requires mechanisms for discovering and correcting errors, as well as providing a feedback of the weaving progress.

## 3 Related Work

Model Weaving, a discipline of Model Engineering, addresses the different types of changes in models, and is commonly used “to unify two complementary, but potentially overlapping, models that describe different views on the same system” [9]. More formally, weaving is a special case of *model transformation*, and it can be expressed in terms of two input models that are operated by a series of steps, and its output is a single new model. Since the first experiments with model weavers made by Bézivin et al. [1], we have seen excellent methods and tools that support this task. We can identify three categories of approximations to model weaving:

### 3.1 General Purpose Modeling Editors

Recent modeling environments and editors of specialized Enterprise Architecture Management tools provide the most simple way of relating model instances through relationships. They provide textual, form-based, and graphical views of the model, which can be used simultaneously for better results. However, these views do not scale when the model is of a considerable size (See Fig. 2), resulting in overlapping relationships, or incomplete views: For instance, Tree editors display relationships as properties of a given element (see Fig. 2), so the modeler has to maintain in his head the current state of the weaving. Also, being a manual task, it is error prone, and the only way of validating the quality of the weaving is by exhaustively looking at each element and relationship.

### 3.2 Specialized Weaving Languages

Considering the pitfalls presented by manual, unconstrained edition of models, several authors have proposed interesting approaches that automate model evolution. Following the Model Driven Paradigm of ‘Everything is a model’, these approaches are supported by an underlying weaving metamodel that provides the syntax for describing concrete weavings between two models.

The ATLAS Model Weaver [4], as well as other approaches such as Epsilon Merging Language [9], offer a textual method using a language for automated matching and merging of changes. These approaches match model elements by using multiple techniques that compute patterns of similarities using different criteria. In particular, element-to-element similarity is often calculated comparing the source and target instance attributes by string comparison and dictionaries of synonyms [4]. As an automated, unsupervised process, this similarity is a coefficient that indicates the level of confidence in the matching.

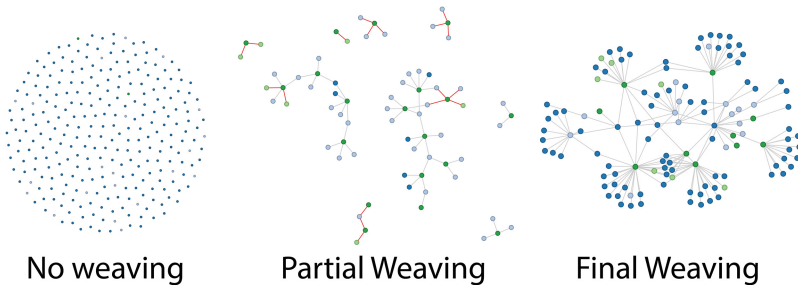
This means that there is always the possibility of having false positives and false negatives in this matching process (see Fig. 1). For this reason, Del Fabro and Valduriez [3] suggest an additional verification stage, thus parametrizing the Weaving Engine to consider cases where none of the pattern matching techniques available could find a high degree of similarity. This requires an exhaustive and manual inspection of both models, as well as validating each proposed match.

### 3.3 Visual Approaches

Fil examines in [5] different methods for the visualization of semantic IS models, and introduces the *business ontology* as a form of integrating multiple metamodels, as it ‘structures the terms and relates the concepts that are used within a business’, and introduces a framework for visualization in IT-based Management. Fil also provides in [6] a method for annotating Visual Modeling languages, thus allowing traceability in the decisions made in the conceptualization phase of modeling.

### 3.4 Outlook

While with these approximations we can perform the weaving of any type of input models, we argue that when dealing with multi-domain models, the burden of guaranteeing coherence is mostly in the head of the modeler. This is an issue caused by their large scale, the high number of possible matches for each inter-domain relationship, and the lack of methods for automatically matching elements with absolute confidence. For this reason, but without seeking to replace these approaches, we propose an additional layer on top of the facilities that these approaches suggest, in order to overcome the cognitive burden and complexity of this task, as well as offering the possibility of inspecting the correctness of this weaving at the same time that it is performed.



**Fig. 3.** Overview of the visual weaving process for an individual relationship type.

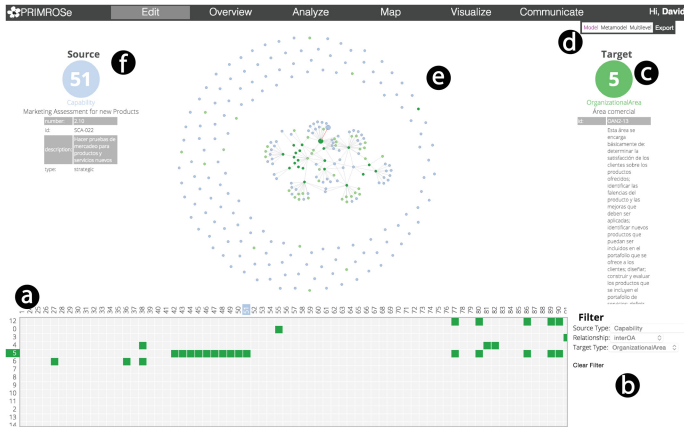
## 4 *Sigourney*: a Visual, Fine-Grained Weaver

Fine-grained Weaving is a goal-oriented and incremental process (see Fig. 3), where the modeler matches pairs of instances for each inter-domain relation type, one pair at the time, and with a refinement and validation stage at the end of each iteration.

We relate this composition to the activity of assembling a jigsaw puzzle. As with real-world puzzles, a good strategy is to first connect pairs of pieces, and when a pattern arises, we can start assembling larger chunks, in order to provide a total view of a fragmented picture. Additional pieces (domains) can be connected to the puzzle (the BM), given that their shapes connect (i.e. concepts between domains are connected by inter-domain relations). A visual approach can be a more effective vehicle for this task, as it does not rely on the memory of the modeler (or his endurance) when adding hundreds (or thousands) of relationships between domains.

### 4.1 Tool Overview

*Sigourney*, our tool for Fine-grained Model Weaving, is supported by PRIM-ROSE, a Visual Analysis platform for Enterprise Models [10]. The workbench of



**Fig. 4.** Workbench of the tool, comprised of multiple sections, which are labeled counter-clockwise: **(a)** Adjacency matrix for a relation type. **(b)** Filter panel. **(c)** Target instance attributes. **(d)** Visualization technique selection. **(e)** Visualization panel. **(f)** Source instance attributes.

*Sigourney* (see Fig. 4) is comprised of two main parts: an Edition Section –at the bottom of the workbench–, and a View Section, displaying a Visualization of the model, as well as detailed information on source and target instances.

## 4.2 Edition

In order to filter the model and show only relevant instances and relationships, this panel contains an adjacency matrix that permits their insertion or removal for given Source and Target metatypes. This matrix is a cross product of all the instances of a Source Type  $S$  versus all the instances of the Target Type  $T$ , so each position in the matrix represents a relationship from an instance of type  $S$  to an instance of type  $T$ .

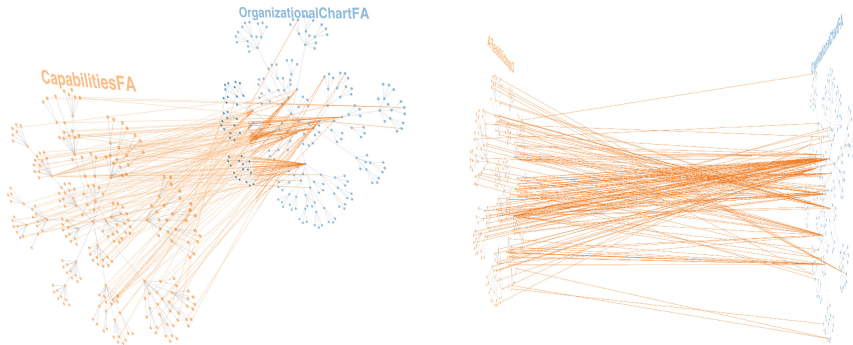
Each modification in the matrix is updated immediately in the Visualization Panel, and hovering on each position of the matrix updates the Source and Target Panels. In order to generate the adjacency matrix, the user must select a Source Type, a relationship name, and a Target Type on the Filter Panel. This selection also updates the Visualization, filtering unrelated elements and relationships.

## 4.3 Visualization

The process of weaving instances of a given pair of metatypes usually starts with an empty adjacency matrix, which translates to a set of unconnected elements (see Fig. 3). As the modeler updates the matrix and connects instances, the visualization is updated with the respective links. The process ends when the modeler deems necessary, and validating the correctness of relationships is easily done, as the visualization contains only the relevant elements.



**Visualization Techniques.** Currently, the tool provides two interactive visualizations for Fine-grained weaving. The user can switch between them at any time, and both favor different purposes. The first technique, displayed in Fig. 4, is a Force-directed Graph, where each color represents a metatype. Unconnected nodes roam freely, while interconnected nodes pull their neighbors a distance proportional to the number of incoming and outgoing links of each node. At any time, the user can switch between an overview of the whole woven model and a filtered model containing relevant instances.



**Fig. 5.** 3D Layered visualization technique. Each point in the plane represents an instance of a concept, and each line parallel to the plane is a domain relationship. The lines between planes are inter-domain relationships.

The second visualization technique is what we call a Layered Visualization (see Fig. 5). While the graph visualization is constrained to a 2-dimensional space, this one allows the user to roam freely and navigate the model as he pleases. This technique consists of two planes, each one containing the force-directed graph of a domain, and inter-domain relationships appear in the midst of both planes, and are relative to the position of the source and target elements. This visualization is useful for assessing the completeness of the weaving between two metatypes, and also allows individual selection of elements and relationships, both intra-domain (in the domain plane) and inter-domain.

## 5 Illustration

Forever Alone (FA) is an *‘Elite’* social networking service for people with similar interests to meet, communicate, and if both persons agree, plan encounters on the real world. These encounters usually are in establishments that have an agreement with Forever Alone, and are recommended by the platform based on the shared interests of both persons. The service is location-aware, and has a recommendation engine for meeting new people near the user.

The target audience of the service is single males and females, with high income margins, over 27 years old, and in general, that are not interested in

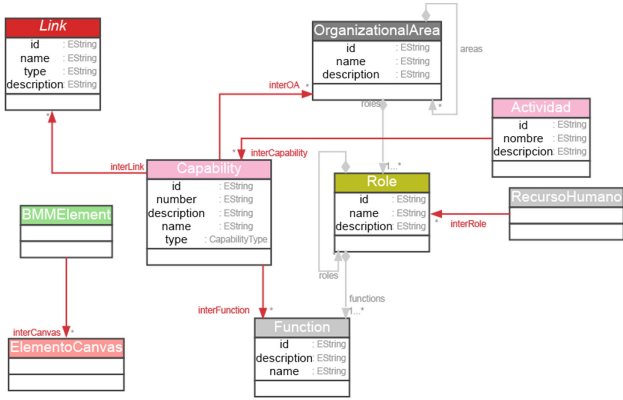


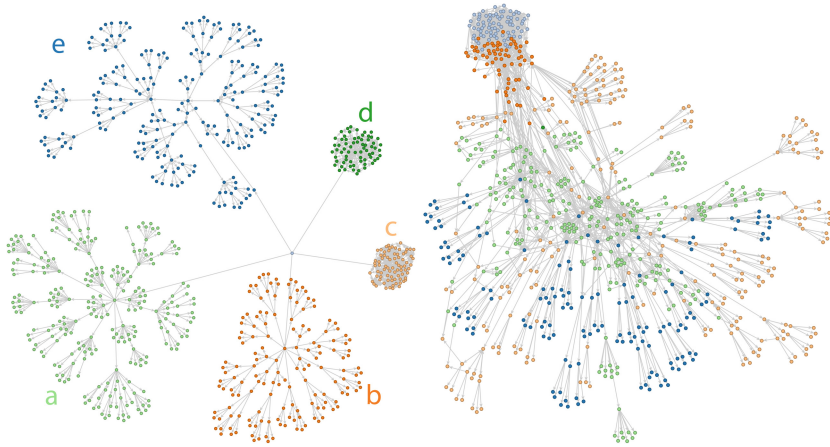
Fig. 6. Fragment of the weaving of Forever Alone at the metamodel level.

having a long-term relationship with anyone. Their high profile usually means that they require additional security and privacy considerations. Being an elite service, Forever Alone is strict with the application process, where potential new users are screened and profiled, in order to determine if they meet the above requirements.

From its inception, the Business Model of Forever Alone was designed using the **Business Model Canvas**, which describes the aspects described above. Its strategy has been described in terms of the **Business Motivation Model** [11], and its high-level **Business Process Architecture** is defined in terms of the Value Chain of the enterprise, using a custom metamodel, where processes are divided into Value-Add (Core), Strategic, and Support Processes. In addition, in order to support decision making and provide a link between the different Business Domains, FA has constructed a *Capability Model*, where high level capacities of the enterprise are formulated, and are realized in the Business Model as Value Chain Links, Organizational Units, and Role Functions. Finally, we have a tailored **Organizational Structure Model** that describes the Areas, Roles, and Functions of employees of the company. The consolidated model contains 798 elements and 1460 relations.

Conscious of the potential gain in knowledge of having an Enterprise Model, for instance, through its analysis, Forever Alone has decided to integrate the Business Domains described above. Each domain model is a detailed description of some aspect of the Business, and it can be considered complex enough.

As a first step, the Enterprise Architecture Committee of FA has integrated five metamodels into one Business Domain Metamodel by connecting concrete concepts of each domain with inter-domain relationships (see Fig. 6). We started by connecting *Capabilities* -from the Capability Model- with *Organizational Areas* of the company. Then, after this weaving is done, we proceeded to connect *Capabilities* with *Value Chain Links*. Later, we matched *Capabilities* with *Functions* that are assigned to Roles of the Organization Structure.



**Fig. 7.** Graph representations of the model before and after the weaving. Circles are model elements, and arcs are relationships between these elements. **Left:** Initial Business Model of Forever Alone, with five different domain models: (a) Capabilities, (b) Business Process Architecture, (c) Business Model Canvas, (d) Business Motivation Model, and (e) Organization Structure. **Right:** Woven Business Model.

Then, we connected **Activities** of the BMC with Capabilities, *Human Resources*, also of the BMC, with *Roles* from the Organization Structure. Finally, we connected Business Motivation Model elements with Canvas Elements. The final result can be seen in Fig. 7, and consists of 798 elements and 2015 relations.

## 6 Conclusion

We have described in this paper a novel way for combining Business Models, by using a special case of Model Weaving, which we call Fine-grained Weaving, and consists of interconnecting these domains with optional relationships at the metamodel, and using a tool of our own, Sigourney, for weaving model instances.

The reasons behind creating this tool, instead of using existing approaches, lie in non-functional aspects of modeling and weaving. We consider that the scale of the task has a negative impact on the modeler, and brings the uncertainty with respect to the coherence of created relationships. In our experience, this integration is difficult to attain, and demands new methods for this case of weaving.

Another benefit of this approach, while not shown on this paper, is that we can have it on top of existing weaving methods, thus serving as a validation and diagnosis tool, something that existing approaches lack.

## References

1. Bézivin, J., Jouault, F., Valduriez, P.: First experiments with a modelweaver. In: Proceedings of the OOPSLA/GPCE: Best Practices for Model-Driven Software

- Development Workshop, 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (2004)
2. Chesbrough, H.: Business model innovation: opportunities and barriers. *Long Range Plan. Bus. Models.* **43**(23), 354–363 (2010)
  3. Del Fabro, M.D., Valduriez, P.: Semi-automatic model integration using matching transformations and weaving models. In: *Proceedings of the 2007 ACM Symposium on Applied Computing, SAC 2007*, pp. 963–970. ACM, New York (2007)
  4. Didonet Del Fabro, M., Valduriez, P.: Towards the efficient development of model transformations using model weaving and matching transformations. *Softw. Syst. Model.* **8**(3), 305–324 (2009)
  5. Fill, H.G.: *Visualisation for Semantic Information Systems. Business and Economics.* Gabler Verlag, Wiesbaden (2009)
  6. Fill, H.-G.: On the conceptualization of a modeling language for semantic model annotations. In: Salinesi, C., Pastor, O. (eds.) *CAiSE Workshops 2011. LNBIP*, vol. 83, pp. 134–148. Springer, Heidelberg (2011)
  7. Frank, U.: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. *Softw. Syst. Model.* **13**(3), 941–962 (2014)
  8. Iacob, M., Meertens, L., Jonkers, H., Quartel, D., Nieuwenhuis, L., van Sinderen, M.: From enterprise architecture to business models and back. *Softw. Syst. Model.* **13**(3), 1059–1083 (2014)
  9. Kolovos, D., Rose, L., Paige, R., Garcia-Dominguez, A.: The epsilon book. *Structure* **178**, 1–10 (2010)
  10. Naranjo, D., Sánchez, M.E., Villalobos, J.: Primrose - a tool for enterprise architecture analysis and diagnosis. In: *ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems, Lisbon, Portugal, 27–30 April, 2014*, vol. 3, pp. 201–213. SciTePress (2014)
  11. OMG: *Business motivation model* (2008)
  12. Osterwalder, A., Pigneur, Y.: *Business Model Generation: A Handbook For Visionaries, Game Changers.* Wiley, Hoboken (2010)
  13. Osterwalder, A., Pigneur, Y., Tucci, C.L.: Clarifying business models: origins, present, and future of the concept. *Commun. Assoc. Inf. Syst.* **16**(1), 1 (2005)
  14. Petrikina, J., Drews, P., Schirmer, I., Zimmermann, K.: Integrating business models and enterprise architecture. In: *18th IEEE International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOC Workshops 2014, Ulm, Germany, September 1–2*, pp. 47–56. IEEE (2014)
  15. Solaimani, S.: *The alignment of business model and business operations within networked enterprise environments.* Ph.D. thesis, Delft University of Technology (2014)
  16. Zott, C., Amit, R., Massa, L.: The business model recent developments and future research. *J. Manage.* **37**, 1019–1042 (2011)