The Devil in the Details: Fine-grained Enterprise Model Weaving

David Naranjo, 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*

Keywords: Enterprise Architecture, Enterprise Models, Business Models, Composition.

Abstract: When building Enterprise Models, it is common to aggregate information from several partial models that describe a fragment of a domain of the enterprise. This integration is made by connecting elements from different domain models, and is usually a manual task, mainly for two reasons: 1) the criteria for connecting pairs of elements is mostly subjective and requires specialized domain knowledge, and 2) because an error may impact the coherency of the whole Enterprise Model. However, manual weaving is a difficult and tedious task, given its complexity and the limited support both from methodologies and tools alike. In this paper, we describe an approach for alleviating the burden on the modeler(s), and consists of connecting model elements by using adjacency matrices, and visualizing changes at the moment they are made, using interactive, coordinated visualizations of the model.

1 INTRODUCTION

Enterprise Models, seen as all the collected information from an enterprise (Naranjo et al., 2013), 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 (tough in some cases extensible with profiles) metamodel, and 2) More recent, customizable approaches, such as Multi-perspective Enterprise Modeling (Frank, 2014), 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

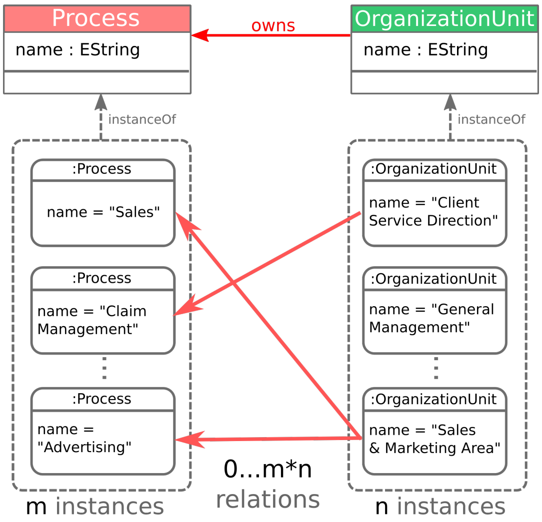

Figure 1: A single mapping of two concepts at the meta-level can result in m*n relationships at the instance level.

now include an ownership relationship to an OrganizationalArea (see Fig. 1)– at the metamodel 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 (such as Epsilon Merging Language, or ATLAS Model Weaver) 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.

We came across this problem when analyzing composite Enterprise Models that contain elements from different domains. In order for multi-domain analyses to make sense, we needed to interconnect related elements. We expected a mostly automatic and smooth weaving of the model, with matching elements being connected by their common attributes. However, we were soon overwhelmed by the large quantity of relationships already present in each domain model, so we could not assess the quality of the weaving, or how many relationships were wrong or missing.

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, Section 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, Section 4 will describe our proposed approach, as well as the tool that implements it. Finally, Section 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

The relationship between the Business and the Operation of an enterprise has been the object of study of several Information Systems disciplines, such as Business Modeling, Business Process Modeling, and Enterprise Architecture, among others. In order to design, implement, and manage software and technological artifacts -as well as to integrate them to the existing organization structure and infrastructure-these disciplines make use of several domain methods that encapsulate the knowledge needed for their practitioners to understand a problem and perform their tasks. 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 problem, that usually is grounded on the business dimension. In fact, *we want* these models and their metamodels to remain small, as it allows us to break a big problem –modeling the enterprise– into manageable parts. After all, modeling is a human activity, and we have problems handling several objects at the same time, i.e. visualizing –grasping– hundreds of components of the model in the same view. In addition, a benefit of using small (meta)models is that we don't have to worry about conformance and coherence *outside* of the scope of the metamodel at hand, something that cannot be done easily with an integrated (i.e. total) Enterprise Metamodel.

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. For instance, we can reuse a subset of the elements described in the BPMN language, in order to have a more simple description of the processes of a company, e.g. for constructing its Value Chain model. This means that when developing specialized domain models, we don't have to start from scratch, as we may have already the information in another model.

With the purpose of exemplifying this integration, we will describe the case of Business Models.

2.1 Business Models

Recently, there has been a lot of interest around the integration of Business Models (BMs) and Enterprise Models (Iacob et al., 2014; Petrikina et al., 2014), 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 (Morris et al., 2005; Al-Debei and Avison, 2010; Casadesus-Masanell and Ricart, 2010; George and Bock, 2011; Zott et al., 2011) 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.

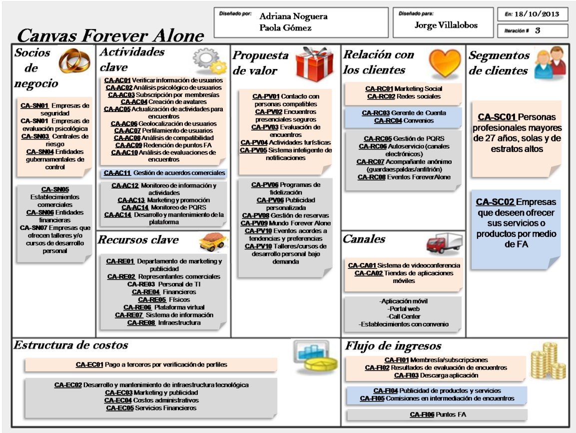

Figure 2: A Business Model Canvas, divided into nine building blocks: Customer Segments (CS), Value Proposition (VP), Channels (CH), Customer Relationships (CR), Key Activities (KA), Key Resources (KR), Key Partnerships (KP), Cost Structure (C$), and Revenue Streams (R$).

For instance, Osterwalder et al. (Osterwalder et al., 2005) 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– (Osterwalder and Pigneur, 2010) (see Fig. 2). 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"*.

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 (Solaimani, 2014), as well as to promote experimentation and innovation with those models (Chesbrough, 2010), e.g. by extending their metamodels with other domains. Moreover, constructs such as Strategy (Magretta, 2002), Ecosystem (Porter et al., 2001), 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.

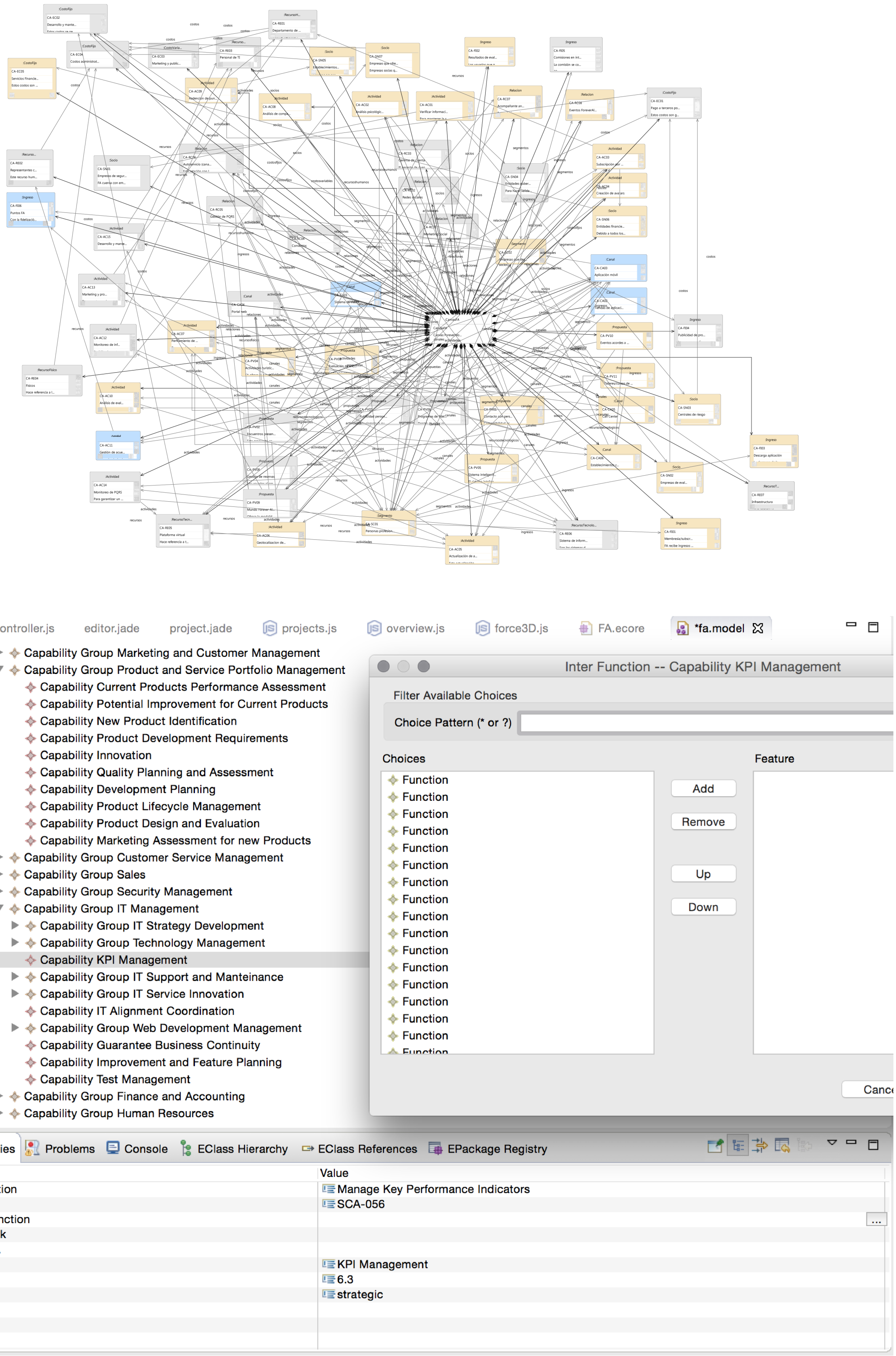

Figure 3: General Purpose Views for editing models. **Top:** A Visual Editor showing a BMC model. **Bottom:** A tree editor of a Business Model.

2.2 *Fine-grained* Weaving

Taking this into account, 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 of 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.

In order to integrate these domains, we make a clear distinction between the multiple domain models and the Consolidated Model (CM) at instance level, as well as the Consolidated MetaModel (CMm) at the meta level. In addition to all the information found on each model, the CM also contains the relationships between elements from different domains. The reason of this distinction is that we need the domain models to evolve in their own ways, as each one may have been developed with a different modeling tool, and with a different modeling method.

The Consolidated Metamodel (CMm) contains the union of all the concepts of every domain, as well as their respective *intra-domain* relationships. We relate pairs of elements from different domains with an optional *inter-domain* relationship between two concepts. At instance level, the CM contains the union of all the domain elements and relations, as well as the inter-domain relations between instances.

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 in cases when there is no apparent connection only by comparing their attributes (see Fig. 1). While this issue does not restrict the use of other weaving methods (see Sec. 3), it demands the possibility of finding mistakes in the weaving process.

In summary, we will use the term *Fine-grained Model Weaving* to refer to the lightweight **integration of different models**, where a series of conditions hold:

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.

Our approach for making this fine-grained weaving 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 by performing Visual Analysis of the model.

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" (Kolovos et al., 2010).

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. (Bézivin et al., 2004), we have seen excellent methods and tools that support this task.

We can identify two categories of approximations to model weaving. First, manual weaving with plain modeling editors, that allow the inclusion of model relations when the domain metamodels have been already woven into an integrated metamodel. As a second category, Specialized Weaving Languages and tools that support weaving with the use of weaving models and languages that allow to express custom rules, depending on the attributes of the respective elements at the instance level.

3.1 General Purpose Modeling Editors

Assuming that we already have a woven metamodel, modeling environments (such as the Eclipse Modeling and Graphical Editing Frameworks), as well as editors of specialized Enterprise Architecture Management tools (e.g. BizzDesign Architect or Sparx Systems Enterprise Architect) provide the most simple way of relating model instances through relationships.

These editors provide textual, form-based, and graphical views of the model, and each we can use multiple of these views simultaneously for better results. However, it is really difficult to perform this weaving using these views when the model is of a considerable size (See Fig. 3), as unrelated elements also appear, occluding relevant relations. On the other hand, we have to consider that each view usually hides some aspect of the model, to improve readability: For instance, Tree editors display relationships as properties of a given element (see Fig. 3), effectively making this weaving more difficult, as the modeler has to maintain in his head the current state of the weaving.

Finally, considering this weaving a manual task, we can argue that 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.

Approaches such as the ATLAS Model Weaver - AMW (Didonet Del Fabro and Valduriez, 2009) provide a graphical method for manual weaving –by using an Eclipse Plugin– that generates the respective weaving model. This method is based on a three-panel Tree view of the weaving model, where it is possible to inspect the properties of source and target elements at the same time, and add weaving rules in between them. However, this editor shares the same problems as General Modeling Editors.

AMW, as well as other approaches such as Epsilon Merging Language -EML (Kolovos et al., 2010), 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 (Didonet Del Fabro and Valduriez, 2009). 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 (Del Fabro and Valduriez, 2007) 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 Outlook

While with these approximations we can perform the weaving of any type of input models, we argue that in the case of Business Models, the burden of guaranteeing coherence in the weaved model is mostly in the head of the modeler. This is an issue caused by the large scale of domain models, the high number of possible matches for each inter-domain relationship, and the lack of objective 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.

4 *Sigourney:* a Visual, Fine-grained Weaver

As with modeling, we propose Fine-grained Weaving as a goal-oriented and incremental process, 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).

We believe that a visual approach for Fine-grained Weaving 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. With this in mind, and complementary to the requirements defined in Section 2.2, we identified a set of requirements for an editor that follows this visual approach:

- **Provide all the details:** As the linking between instances will be made manually by an expert, and his criteria for matching is subjective and based on his domain knowledge, we must offer all the possible information needed for his task.

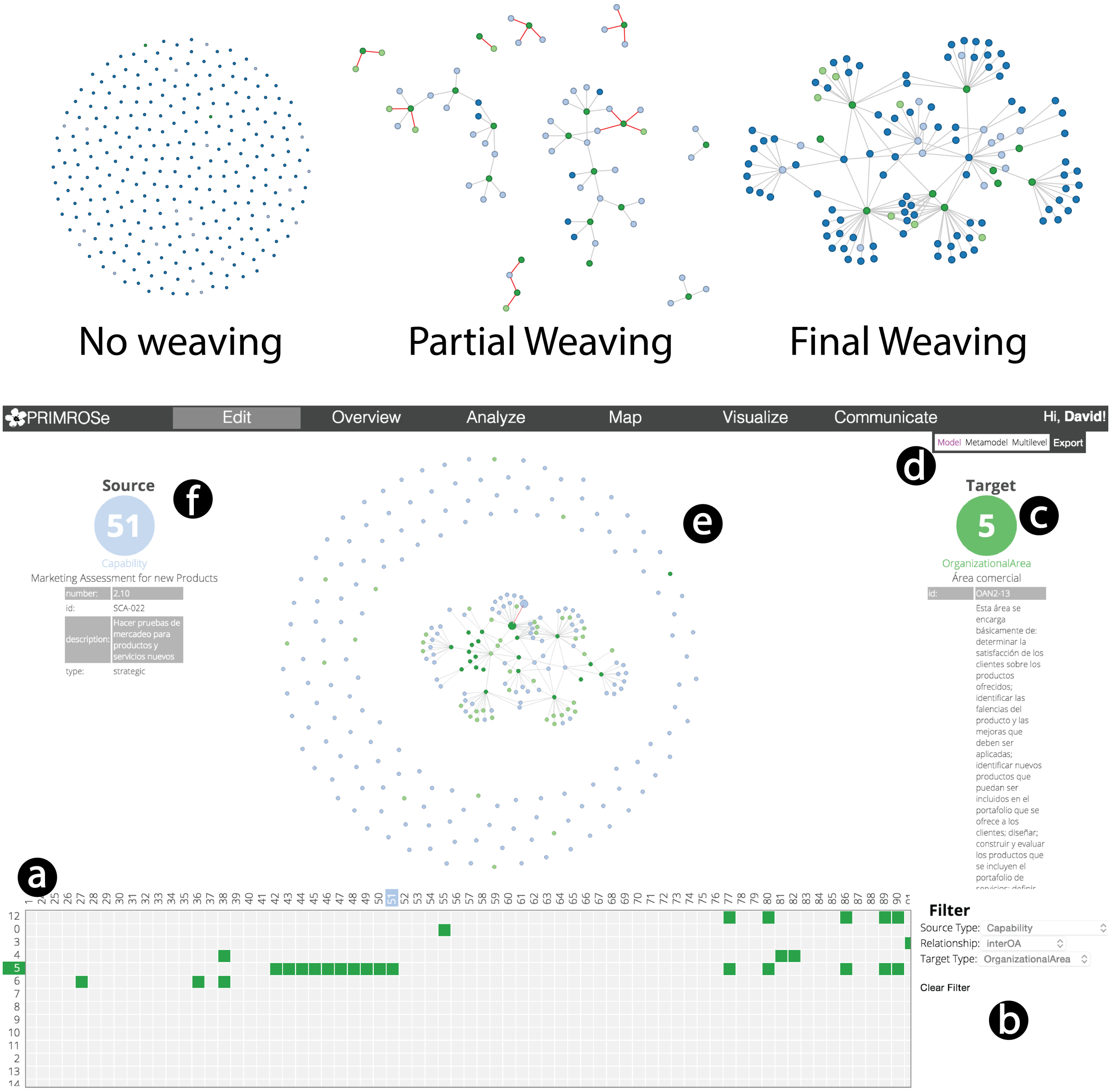

Figure 4: **Top:** Overview of the Visual Weaving process for an individual relationship type. **Bottom:** Workbench of the tool, comprised of multiple sections, which are labeled counter-clockwise: **a)** Adjacency Matrix for a relation type. **b)** Filter panel, where the user selects Source Type, Relation name, and Target Type. **c)** Attributes for target instance for a given relationship on the model. **d)** Visualization Technique Selection, where the user can switch between the *Graph*, *Metamodel*, and *Layered* Visualizations. **e)** Visualization panel, displaying in this case a filtered Graph View. **f)** Source instance attributes.

- **Offer multiple visualization methods:** Every visualization technique has its benefits and drawbacks. Thus, we require more than one way of visualizing the same model, in order to reach complementary insights.
- **Handle different levels of detail:** In order to diagnose and validate correctness, we require two perspectives: 1) A total/holistic view of the whole model, that allows overall pattern recognition in order to find general bad smells and outliers, and 2) A filtered view that helps the modeler in assessing localized issues, and gives perspective on the completeness of the weaving.
- **Avoid Distractions:** Graphical Interfaces of modeling environments provide a multitude of options and tools that won't be needed in this task. We seek a clean interface, with a minimal set of tools and options, to provide focus.
- **Tame Visual Complexity:** The use visualization techniques and good practices for reducing the clutter involved in displaying thousands of elements and relations.
- **Multiple, coordinated views:** To have several complementary views at the same time, where changes in one view affect the others.
- **Continuous and seamless feedback:** The addition of removal of an individual relationship should be visualized immediately. Also, performance and response times of the tool are critical.

4.1 Tool Overview

Sigourney, our tool for Fine-grained Model Weaving, is supported by PRIMROSe, a Visual Analysis platform for Enterprise Models (Naranjo et al., 2014). The workbench of *Sigourney* (see Fig. 4, Bottom) 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

This Section contains an adjacency matrix that permits the insertion or removal of relationships, and selectors for Source and Target metatypes, along with the relationship type, in order to filter the model and show only relevant instances and relations.

The adjacency 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. 4, Top). 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.

4.3.1 Visualization Techniques

Currently, Sigourney provides two interactive visualization techniques 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 neighbours a distance proportional to the number of incoming and outgoing links of each node. At any time, the user can switch between a total graph that is an overview of the whole woven model, and a filtered graph, that is a sub-model containing relevant instances. The topology of this graph does not differentiate different domains.

The second visualization technique is what we call a Layered Visualization (see Fig. 7). While the graph visualization is constrained to a 2-dimensional space, this one has 3 dimensions, allowing 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.

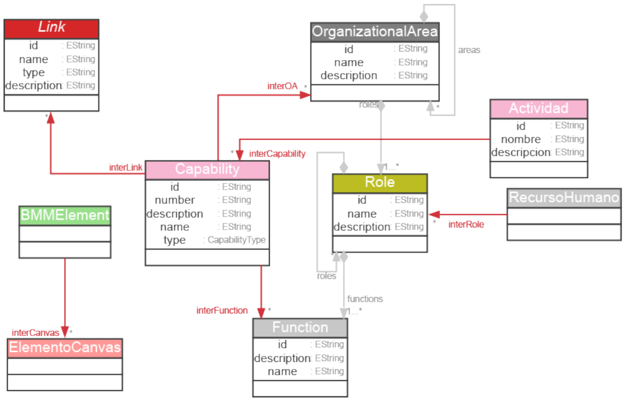

Figure 5: Fragment of the weaving of Forever Alone at the metamodel level.

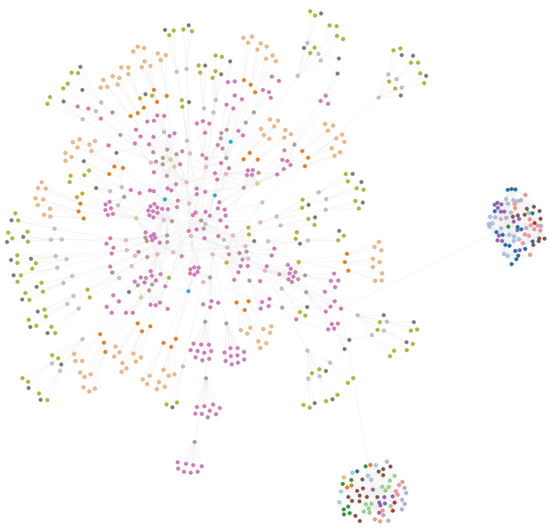

Figure 6: Partial weaving of three Business Domains at the instance level.

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 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** (OMG, 2008), 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 Domain, and it can be considered complex enough.

As a first step, the Enterprise Architecture Committee of FA has integrated the five metamodels into one Business Domain Metamodel by connecting concrete concepts of each domain with six inter-domain relationships (see Fig. 5).

At the beginning, as we can see in the top of Figure 8, the five domains are not connected between them. In order to weave this model, 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. At this point, we can observe three of the domains woven into one –see Fig. 6).

Then, we connected **Activities** of the BMC with Capabilities, *Human Resources*, also of the BMC, with *Roles* from the Organization Structure. Finally, we weaved Business Motivation Model elements with Canvas Elements. The final result can be seen in Fig. 8, Bottom, and resulted in 798 elements and 2015 relations.

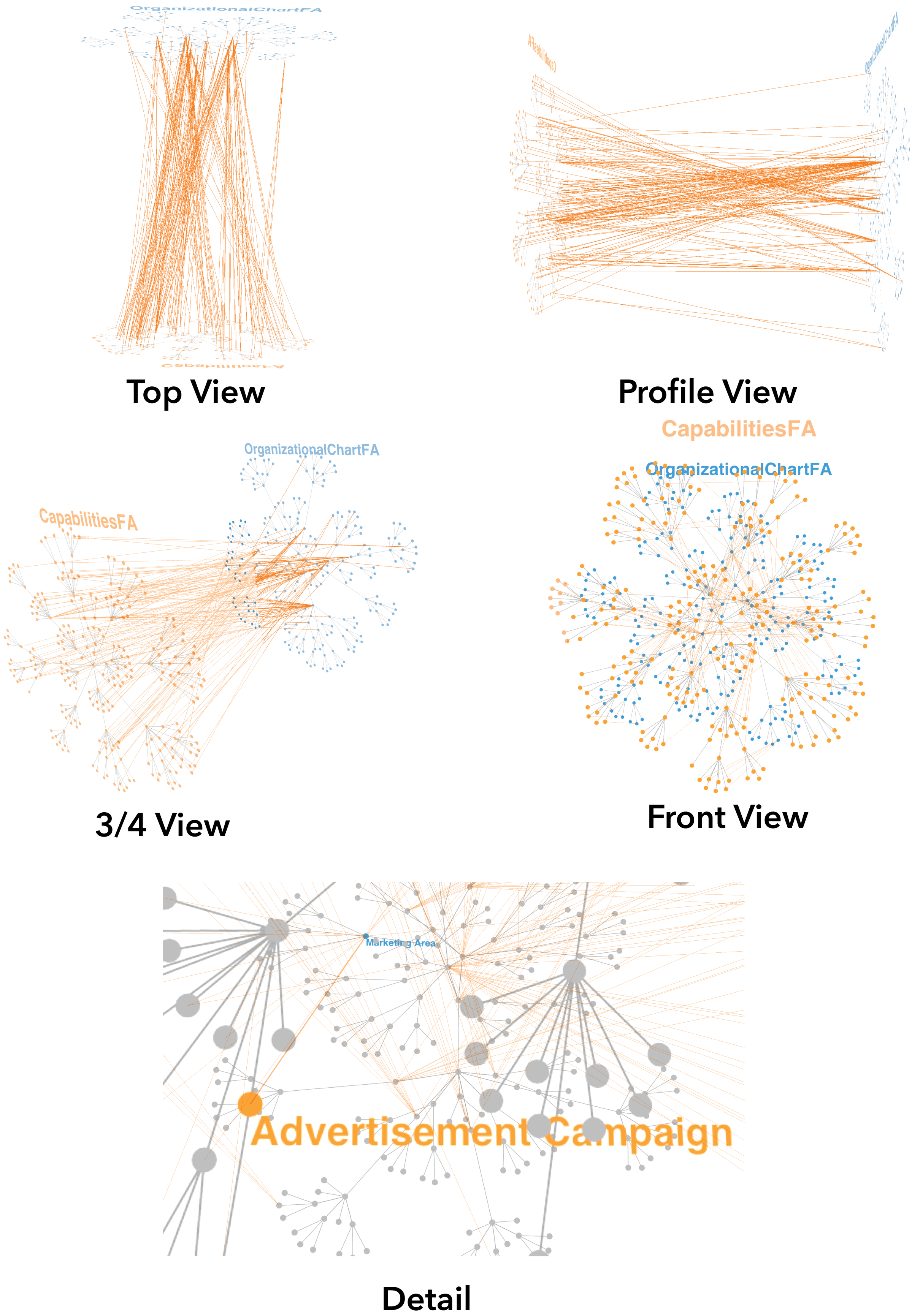

Figure 7: Multiple views of the Layered Visualization Technique.

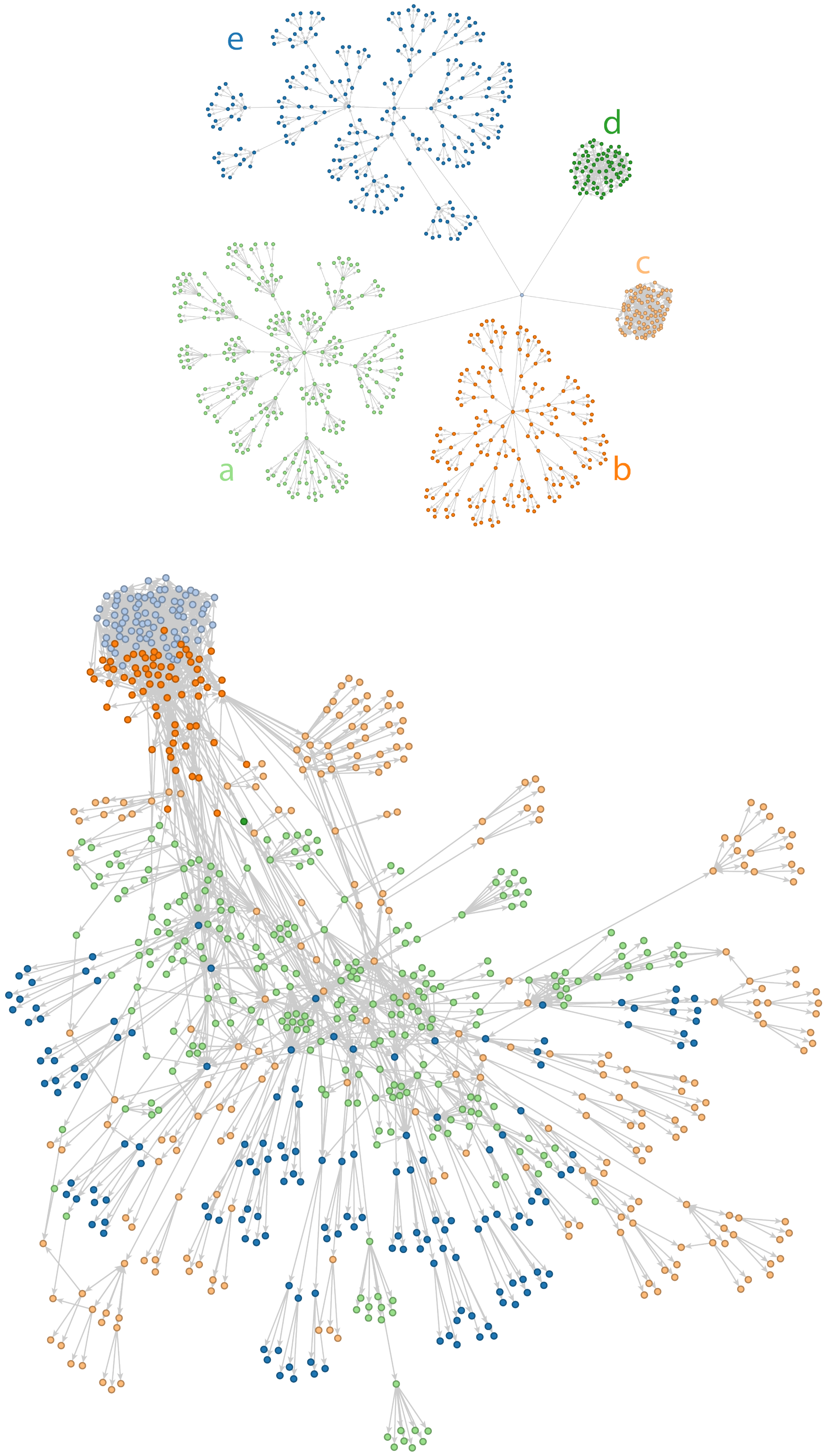

Figure 8: **Top:** Initial Business Model of Forever Alone, with five different domain models: a)Capabilities, b)Business Process Architecture, c) Business Model Canvas, d)Business Motivation Model, and e) Organization Structure. **Bottom:**Weaved Business Model.

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 a tool of our own, instead of using existing approaches, lie in non-functional aspects of modeling and weaving. We consider that the scale of the task impacts negatively on the modeler, as well as the uncertainty with respect to the coherence of created relations. 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

Al-Debei, M. M. and Avison, D. (2010). Developing a unified framework of the business model concept. *European Journal of Information Systems*, 19(3):359–376.

Bézivin, J., Jouault, F., and Valduriez, P. (2004). 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*.

Casadesus-Masanell, R. and Ricart, J. E. (2010). From strategy to business models and onto tactics. *Long Range Planning*, 43(23):195 – 215. Business Models.

Chesbrough, H. (2010). Business model innovation: Opportunities and barriers. *Long Range Planning*, 43(23):354 – 363. Business Models.

Del Fabro, M. D. and Valduriez, P. (2007). Semi-automatic model integration using matching transformations and weaving models. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 963–970, New York, NY, USA. ACM.

Didonet Del Fabro, M. and Valduriez, P. (2009). Towards the efficient development of model transformations using model weaving and matching transformations. *Software & Systems Modeling*, 8(3):305–324.

Frank, U. (2014). Multi-perspective enterprise modeling: Foundational concepts, prospects and future research challenges. *Softw. Syst. Model.*, 13(3):941–962.

George, G. and Bock, A. J. (2011). The business model in practice and its implications for entrepreneurship research. *Entrepreneurship Theory and Practice*, 35(1):83–111.

Iacob, M., Meertens, L., Jonkers, H., Quartel, D., Nieuwenhuis, L., and van Sinderen, M. (2014). From enterprise architecture to business models and back. *Software & Systems Modeling*, 13(3):1059–1083.

Kolovos, D., Rose, L., Paige, R., and Garcıa-Domınguez, A. (2010). The epsilon book. *Structure*, 178:1–10.

Magretta, J. (2002). Why Business Models Matter. *Harvard Business Review*, 80(5):86–92.

Morris, M., Schindehutte, M., and Allen, J. (2005). The entrepreneur's business model: toward a unified perspective. *Journal of Business Research*, 58(6):726 – 735. Special Section: The Nonprofit Marketing Landscape.

Naranjo, D., Sánchez, M. E., and Villalobos, J. (2013). Connecting the dots: Examining visualization techniques for enterprise architecture model analysis. In Grabis, J., Kirikova, M., Zdravkovic, J., and Stirna, J., editors, *Short Paper Proceedings of the 6th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2013), Riga, Latvia, November 6-7, 2013.*, volume 1023 of *CEUR Workshop Proceedings*, pages 29–38. CEUR-WS.org.

Naranjo, D., Sánchez, M. E., and Villalobos, J. (2014). Primrose - A tool for enterprise architecture analysis and diagnosis. In Hammoudi, S., Maciaszek, L. A., and Cordeiro, J., editors, *ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems, Volume 3, Lisbon, Portugal, 27-30 April, 2014*, pages 201–213. SciTePress.

Naranjo, D., Sánchez, M. E., and Villalobos, J. (2015). Evaluating the capabilities of enterprise architecture modeling tools for visual analysis. *Journal of Object Technology*, 14(1).

OMG (2008). Business motivation model.

Osterwalder, A. and Pigneur, Y. (2010). *Business Model Generation: A Handbook For Visionaries, Game Changers, And Challengers*. Wiley.

Osterwalder, A., Pigneur, Y., and Tucci, C. L. (2005). Clarifying business models: Origins, present, and future of the concept. *Communications of the association for Information Systems*, 16(1):1.

Petrikina, J., Drews, P., Schirmer, I., and Zimmermann, K. (2014). Integrating business models and enterprise architecture. In Grossmann, G., Hallé, S., Karastoyanova, D., Reichert, M., and Rinderle-Ma, S., editors, *18th IEEE International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOC Workshops 2014, Ulm, Germany, September 1-2, 2014*, pages 47–56. IEEE.

Porter, M. E. et al. (2001). Strategy and the internet. *Harvard business review*, 79(3):62–79.

Solaimani, S. (2014). *The alignment of Business Model and Business Operations within Networked Enterprise Environments*. PhD thesis, Delft University of Technology.

Zott, C., Amit, R., and Massa, L. (2011). The business model: Recent developments and future research. *Journal of Management*.