Executable Business Model Blueprints

Completed Research

María Camila Romero
Systems and Computing Engineering,
Universidad de los Andes, Bogotá,
Colombia
mc.romero578@uniandes.edu.co

Mario Sánchez
Systems and Computing Engineering,
Universidad de los Andes, Bogotá,
Colombia
mar-san1@uniandes.edu.co

Jorge Villalobos
Systems and Computing Engineering,
Universidad de los Andes, Bogotá, Colombia
jvillalo@uniandes.edu.co

Abstract

The business architecture shapes the Enterprise Architecture as it describes the enterprise in both its intentional and operational aspects. While the operational dimension can be perceived in the enterprise’s daily activities, intentional aspects are more abstract and difficult to understand. One of these aspects is the business model, which defines how an enterprise generates value. Therefore, this model is crucial to understand the other components in an enterprise architecture, from the business strategy to the operational model. In spite of this, most approaches to analyze a business model focus on static aspects, providing a disconnected view of its components and making it difficult to perform in-depth analysis. To address this situation, this paper proposes an executable business model blueprint to represent the business’ structural and dynamic aspects. This enables dynamic analyses based on simulation techniques to reveal hidden business’ dynamics and gain a further understanding of the overall business behavior.

Keywords

Business Model, Model Driven Architecture, Dynamic Analysis.

Introduction

The comprehension of the business architecture is an essential requirement for the development of an enterprise architecture (TOGAF 2009). This understanding is enabled by means of models, which provide representations of different business dimensions, and become inputs for analysis. One of these models is the business model, which depicts the way in which a business transforms value, information and money. This model is thus the key to understand what a business does, as it describes the components that shape the business behavior: in a coherent business architecture, the existence of all the elements found in other models should be somehow justified by business model components.

In the last two decades, several approaches and theories have been formulated to enhance the comprehension and use of the business model (Zott et al. 2011). Nonetheless, most of them have focused on elements to describe the structure of the business leaving most of the dynamic aspects to intuition. There have been attempts to portray the dynamics behind this structure, but this has proven difficult because representing both aspects requires a high effort and typically results in very complex models.

In order to understand the way in which a business works, it is necessary to understand its dynamics, which is hard to do only through static analysis. Dynamic analysis techniques are required to understand how components in a business relate to each other over time, to study the impact of non-controllable internal or external factors, and how actions on some of them have repercussions over the rest of the system. Dynamic analysis can also be used to test hypothesis and help to predict specific outcomes, thus helping the diagnose of businesses in their early stages or during transformation phases.
Given the lack of means to describe and analyze the dynamic aspects of a business, and the importance to so in the context of enterprise architecture, this paper proposes an executable business model blueprint that combines the structural and dynamic components and enables their study under the influence of non-controllable variables. This blueprint was created after analyzing existing approaches to describe business models and identifying the fundamental building blocks required to describe a business model. These were complemented with details about the relations between them and their behavior. As a result, the blueprint can be used to perform static and structural analyses, as well as dynamic analyses that contribute to a better understanding of the business.

The structure of this paper is as follows. Section 2 presents an overview of the existing approaches that we studied to create our proposal. This is followed, in Section 3, with the presentation of the 8 building blocks to describe a business model. These blocks are made concrete in the business model blueprint: Section 4 describes its static elements, while Section 5 focuses on the dynamic components. The usage of the blueprint for modeling and analysis is illustrated with a case study simple enough to fit in the available space. Finally, Section 6 concludes our paper.

**Business Model Definitions**

The essence of any enterprise lies in the way in which it acquires and transforms resources, and delivers value to its clients, typically in exchange for money. When an enterprise is stripped of operational aspects, such as processes, people, and machines, what remains is the business; and even though this is a reduced view of an enterprise, it remains highly complex due to the high number of elements and relationships involved. The term business model has been used to refer to these elementary views of enterprises, which are used as a mean to manage complexity during the study and design of businesses.

<table>
<thead>
<tr>
<th>APPROACH</th>
<th>DEFINITIONS</th>
<th>CONCEPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Gordijn and Akkermans 2001)</td>
<td>How value is created and exchanged within the stakeholder network [...] the way of doing business between actors.</td>
<td>Actor, Value Object, Value Port, Value Interface, Value Exchange, Value Offering, Market Segment, Composite Actor, Value Activity</td>
</tr>
<tr>
<td>(Magretta 2002)</td>
<td>Stories that explain how enterprises work</td>
<td>Customer, Value, Money, Activities (Making and Selling)</td>
</tr>
<tr>
<td>(Linder and Cantrell. 2000)</td>
<td>Organizations’ core logic for creating value</td>
<td>Revenue Sources, Value Proposition, Delivery Model, Funding Model, Assets, Capabilities, Relationships, Knowledge, Customers</td>
</tr>
<tr>
<td>(Rappa 2001)</td>
<td>Method of doing business by which a company can sustain itself – that is, generate revenue.</td>
<td>Brokers, Buyers, Sellers, Transactions, Broadcaster (Web Page), Services, Data, Consumers, Infomediaries, Retailers, Manufacturer, Affiliates, Revenue</td>
</tr>
</tbody>
</table>

**Table 1 Business Model Approaches**

Existing attempts to understand and characterize what a business model is have resulted in different sets of fundamental concepts and tools. E3-Value (Gordijn and Akkermans 2001) and Osterwalder’s canvas (Osterwalder and Pigneur 2010) are just two examples which have resulted in concepts that appear very different. There have also been differences in methodology, such as taxonomy-based approaches (Linder and Cantrell. 2000), (Rappa 2001), approaches based on general descriptions (Magretta 2002) and approaches that have contributed visual notations (Gordijn and Akkermans 2001), (Osterwalder and Pigneur 2010). Table 1 presents a summary of these approaches, highlighting their definition of business model and their set of fundamental components.
A common characteristic between these approaches is their focus on the static elements that constitute the backbone of a business model. That is, they identify the components that are part of a business model and the structural relations between them. They are thus useful to describe business models from a structural point of view and to support analysis based on said structures. For example, in the case of Osterwalder’s canvas it is possible to study the dependencies between activities, resources and value propositions in relation to customer segments.

On the other hand, fewer approaches have attempted to represent and analyze the dynamic aspects of a business model. Most of them have enriched static representations with dynamic components to test them under dynamic scenarios. In particular, they have focused in the analysis of the relations among structural elements and the influence that they have in one another. However, it is not yet clear what the dynamics of a business model are and what are the basic dynamic concepts that should be described.

Existing approaches have also applied known techniques and tools to the business model domain. In the case of (Duersch 1975), a simulation framework based on behavioral modeling and experience record was defined. Within this framework, business’ dynamics is described in terms of three relations: market share performance, net income and investment, and performance measures. These relations are analyzed using historical records and observed experiences, and used to run simulations. Ultimately, these enable a better understanding of the relations by presenting their behavior as time passes by. A related approach was presented in (Schmitt et al. 2005), where a simulation tool based on ebXML transactions was used to study the dynamics of e-businesses and to compare business models. System dynamics was used in (Onggo et al. 2006) to analyze the behavior of business models.

**Business Model Building Blocks**

Models are abstractions and representations of systems under study, built with a specification or descriptive purpose (Favre 2005). In this sense, a business model is an abstraction of a business which is built to describe and understand a business. However, to achieve an adequate understanding and guarantee an accurate representation, the business model should be faithful to the business. This depends strongly on the modeling language: if it lacks the expressiveness to represent the business, the modeling process will never deliver a useful business model.

Past initiatives to define business models have resulted in different modeling languages and modeling approaches. Although they have focused in the structural aspects, they have identified different sets of key elements, hence making it difficult to uniquely answer the question “What constitutes a business model?”. In the rest of this section we present the result of our work to identify in these works a base set of concepts to represent business models, which we have called the business model building blocks.

The strategy that we used had as main goal to identify clusters of concepts in the existing approaches. For this, we started with the concepts presented in Table 1 and followed a process inspired by (Babur et al. 2016). We carefully studied each concept, its definition in the context of business models, and its meaning in the English language. Even if it seems as if the concepts in each approach are uniquely defined, by analyzing their meaning it was possible to identify similar components: two components are similar because they have the same meaning or because they are contained in each other. Similar components constitute a cluster. Furthermore, to establish if a concept was contained in another concept, we used the Wordnet tool (Feinerer and Hornik 2006) and used hyponyms to determine if there was a super-subordinate relation.

The clustering process resulted in the grouping presented in Table 2. The table presents the cluster name, the concepts grouped, and the reason why they were placed in the cluster: S or R. An S stands for shared term which means that the component is a synonym of the cluster’s name, or they both have the same name. R stands for a hyponym relation with the cluster’s name. In some cases, the name of the cluster or component is accompanied by the term from with the direct hyponym relation was established.

<table>
<thead>
<tr>
<th>CLUSTER</th>
<th>COMPONENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions (R:Act)</td>
<td>Activities (R), Value Exchange (R), Value Activity(R),</td>
</tr>
<tr>
<td></td>
<td>Activities(R), Delivery Model (R), Funding Model (R),</td>
</tr>
<tr>
<td></td>
<td>Transaction (R).</td>
</tr>
</tbody>
</table>
Based on these clusters, we identified 8 building blocks that constitute the base for the business models:

- **Actions**: Activities performed by agents.
- **Agent**: Someone or something that performs actions that enable exchanges.
- **Value Flow**: Goods or services that are exchanged between agents.
- **Money Flow**: Money that is paid for or received in exchange for value.
- **Information Flow**: Messages that are exchanged to enable the exchange of value and money.
- **Ports**: Interfaces that define the origins and destinations of flows.
- **Relationship**: A connection between two or more agents.
- **Resource**: Elements that support the execution of actions.

The structure of a business model is thus formed by agents that perform actions and by resources that support the execution of actions. Agents exchange money, information, or value. Since these exchanges happen constantly, it is possible to perceive them as streams or flows. Consequently, relations among structural elements can be distinguished as flows characterized by the element that flows (e.g., value flow). Flows define dependencies among components, and enable transformation processes.

The dynamics of the business is defined in part by the flows and in part by the actions performed by agents. An action is product of the initiative of an agent and triggers flows by transforming inputs into outputs. Flows connect agents through their ports and actions are triggered in response to these flows. Therefore, a cause-effect relationship can be established between an agent that performs an action, and the one that receives the output of the transformation.

### Static Blueprint of a Business Model

In order to use the building blocks that we identified as the foundation for creating business models, we created a modeling language. This language was defined based on the metamodel shown in Figure 1, and it fully incorporates the building blocks, albeit with a few name changes. The rest of this section maps the building blocks to the metamodel concepts and then exemplifies how it can be used to describe the static aspects of a business model blueprint.

#### Table 2 Business Components Cluster

<table>
<thead>
<tr>
<th>Component</th>
<th>Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent (R: Causal Agent)</td>
<td>Partners (R), Customer Segments (R), Actor(S), Market Segment (Actors, S), Composite Actor(S), Brokers (R), Buyers (R), Sellers (R), Broadcaster (R), Consumers (R), Infomediaries (Intermediary (R)), Retailers (R), Manufacturers (Maker, R), Affiliates (R), Customer (R), Customers(R).</td>
</tr>
<tr>
<td>Value (Goods or Services)</td>
<td>Value Proposition (S), Value Object (S), Value(S), Value Proposition(S), Services (S).</td>
</tr>
<tr>
<td>Money (R: Amount of Money)</td>
<td>Revenue Streams (R), Cost Structure, Value Object (Money-S), Money(S), Revenue Sources (R), Cost Structure, Revenue (R)</td>
</tr>
<tr>
<td>Information</td>
<td>Value Object (S), Messages, Data (R), Knowledge (S)</td>
</tr>
<tr>
<td>Ports (R: Communication)</td>
<td>Value Interface (R), Value Port (S), Channels (Interface: R).</td>
</tr>
<tr>
<td>Relationships</td>
<td>Customer Relationships (S), Relationships (S).</td>
</tr>
<tr>
<td>Resources</td>
<td>Resources (S), Assets (R).</td>
</tr>
</tbody>
</table>

Twenty-fourth Americas Conference on Information Systems, New Orleans, 2018
Agents

Agents in a blueprint are referred to as processors, which may have one or more instances. When there is only one instance (e.g., the director of sales), we call it a Single Processor. When there are many instances of the same processor, and it is possible to assume that they all behave in the same way (e.g., salespersons), we refer to the processor as a Composite Processor. Finally, we use Groups to refer to complex combinations of processors and accumulations, such as the sales department of an enterprise.

Resources

Resources relevant in a business model can be raw materials, personnel or physical locations, among others. Depending on the resource type, they may be presented like processors or flows. Furthermore, some resources can be stored. For this reason, the metamodel includes the concept of accumulation, which represents any kind of storage facility for resources, and is characterized by the type of items stored and by its maximum capacity. Some examples of accumulations are warehouses, bank accounts or a list of unsettled orders.

Flows, Ports and Relations

Flows connect processors and accumulations, and they are defined by the type of items that can flow through them. There are three basic types of flow: Information, Value and Money. These types can be specialized into subtypes, depending on the needs of the particular business that is being modeled. Because of the importance of the types of items flowing in the model, we introduce Ports which are attached to processors and control items flowing into and out of them. By characterizing ports with the types and subtypes relevant for the business model, it is possible to characterize and monitor different flows.

There is no direct mapping between the last remaining building block, Relations, and a concept in the metamodel. Instead, Relations are represented by complex networks of elements that include processors, flows and ports.

A Case Study

To illustrate the construction of a business model blueprint, we will use a simplified case study based on a grocery store which sells perishable and nonperishable goods to customers. The goods are acquired from various suppliers by a warehouse manager, who is later in charge of storing them in a warehouse, or placing them in shelves and refrigerators. Customers can buy products directly from the store or use a mobile application to place a delivery order. When buying in the store, a cashier is in charge of receiving the order and the cash for the requested products. When the order is placed as a delivery, the client pays online and the warehouse manager prepares the order and gives it to one of the two delivery guys, which then delivers it to the client.

To facilitate the blueprint’s understandability, we designed a visual language pursuing the benefits of graphic notations (Ottensooser et al. 2012). The language was designed considering the three components of a visual notation, and the design principles for effective visual notations (Moody 2009). The resulting language is illustrated in Figure 2, which depicts the model of the grocery store.
Business Elements and Flows

To represent the grocery, store static business model, we identified the main components of the meta-model taking into account the main business model building blocks. We identified four processors: Suppliers, Store Clients, Delivery Clients and the store itself. To portray them, we used circles. Suppliers and the Store group other business elements and thus they are modeled as groups. To represent groups, we used circles with a heavy stroke border. Since we can assume that all clients have a similar behavior, Clients are modeled as a composite processor. These are represented using double circles.

Based on these business elements we identified the connections between them i.e., flows. Recalling that there are three types of flows, we initially established the Value flows. In general, to define a flow one needs to identify a source (that triggers the flow) and a sink (that receives it). As the Value items that flow are goods, we used a small rectangle to represent the item and a heavy stroked arrow for the flow. The first flow is established between the suppliers and the store (as they deliver the goods that will be sold); the remaining flows are defined between the store and the clients as goods are sold. Since there may be sold in the store or as a delivery order, we represent the delivery as a different value item. Finally, as clients accumulate goods, we identify the correspondent accumulations. In this case, they are represented with a rectangle and connected to the corresponding participant with a value flow.

![Figure 2 Store Model](image)

Details in the Model

Figure 2 presents a global view of the store’ business model that serves to identify the basic components and the value flows. Figure 3 presents a more detailed view, which shows the money and information flows along with the business elements grouped in the processors from Figure 2.

![Figure 3 Grocery Store Detailed View](image)

To represent the money and information flows we followed a similar process as in the value flows and found that there are three money flows between the suppliers and the store, and the clients and the store. With respect to information, we identified two information items: orders, and order confirmations. Every time a business element ordered a value item, an information flow carrying said order was established and was followed by an order confirmation.
Within the Suppliers group, we can see that they have been differentiated between perishable (P.S.) and nonperishable (NP.S.) suppliers. Both are represented using composite processors. The Store group, on the other hand, presents the various actors introduced in the scenario description. It is important to note that elements like the warehouse, shelves and refrigerators are presented as accumulations. Moreover, the app used by clients is portrayed as a single processor as it is the channel used to receive client’s orders.

Additional detail can also be found in the identified flows as they have also been established inside groups: the flows that connect the groups are extended to the corresponding processors in each group in order to clearly depict the source and sink of the connection. For instance, the flow that connected the suppliers and the store was extended to each type of supplier, and to the warehouse manager.

Considering the way in which detail is handled, it is clear that Figure 3 can still be enriched. Still, when adding details to the model (or removing them to get a simplified model) one must take into consideration that consistency between elements must be kept. For instance, to view only the Value flows as in Figure 2, one should present all the processors even when they are not connected by any of these flows (for instance the mobile app).

**Executable Business Model Blueprint**

The previous section discussed the static aspects of a business model blueprint. We now explain the two elements that serve to describe the dynamic aspect of a business model: processors’ behaviors and port’s queues.

**Behavior:** The behavior of a processor defines the particular actions that a processor is able to do at a given time. Hence, it is defined as a set of actions that a processor should do in response to events and conditions. There are three kinds of events of interest for agents: timed events, which are defined in terms of specific time (a date, hour, etc.) or period (e.g., the first day of the month) and flow-arrival events, which occur when something is received through a flow.

Agents’ actions are combinations of simple operations: pick an item from a flow, transform an item, generate a new item, query and update an accumulation, and put an item into a flow. As an example, the behavior of an order-processing agent may be to take an incoming purchase order (information flow), verify if the warehouse (accumulation) has enough stock of the product, send the product (value flow) and invoice (information flow) to the customer, and update the warehouse inventory.

**Queues:** Considering the multiple exchanges that take place within the model, and the fact that processors are not able to react instantaneously to all of them, we define queues to portray how processors manage the reception of flowing items. Queues can be found in entry ports as they are the points to which flow items arrive. They may or may not have a limit considering the business’ logic, and they are managed by the processor who owns the entry port. Thus, queue management is also part of the processor’s behavior and is defined in terms of the processing action. Queues are able to hold flowing items until the processor is able to act upon them. When a flow arrives to a port, the carried items are placed in the queue.

**Dynamics in the Grocery Store**

The dynamic of the business model of the grocery store is defined considering all the business elements presented in Figure 2B. Considering the complexity of the whole example, we will illustrate the dynamics using only the delivery phase (see Figure 4): Once a client places an order through the mobile app, the order is sent to the warehouse manager (1), who picks up the items from the warehouse (2) and gives them to one of the delivery guys (3) for him to take it to the client. Once the order is received by the delivery guy (4), he delivers it to the client taking into account the current orders.

The identifiable behaviors in this scenario correspond to the mobile app, the warehouse manager and the delivery guy. First, the app sends the order to the warehouse manager, who gathers the elements of the order depending on the current delivery order. This behavior can be described as sending an information token (app) and processing the order (warehouse manager). This last action depends on the orders in the manager’s queue. Once the order is ready, the warehouse manager gives it to one of the delivery guys (send value token). This selection will depend in the coverage and availability. If the delivery guy that covers the
area of the client is available he will be given the order; if not, the guy with the closest area will deliver the order (send value token).

![Figure 4 Dynamic View](image)

**Blueprint Execution**

Considering all the static and dynamic elements available to describe a business model blueprint, it is now possible to create a platform capable of running the blueprints and support dynamic analysis. We created an initial version of this platform as a simulation machine, which is capable of loading blueprint definitions and executing them. For this purpose, the machine is capable of generating, keeping track, and distributing all the events of relevance for the blueprint, executing agents’ actions as specified. We chose a discrete event simulation approach where the state of the systems only changes when an event occurs (Banks 1998).

The simulation machine is also capable of loading scenario definitions in order to test hypothesis and perform experiments. During the execution, the simulation machine generates traces of the processors actions and about the accumulation current statuses, which are then stored for analysis and comparison. This simulation core is complemented by a graphical editor for the blueprint and the scenarios, and a tool for visualizing the traces and the results of the executions.

The core of the tool, where blueprints are executed, was developed using the Java language, and was designed to support all kinds of processor behaviors and to be extendable. For this purpose, we defined an interface (IBehavior) that abstracts the common actions that a processor can perform. The interface declares 2 methods intended to specify the specific and periodic time events relevant to the processor. 3 additional methods trigger the exchange of information, value and money tokens, 1 method calls for queue checks and the remaining 2 establish the relation between events and actions. Simple processors can be created by configuration, using one of the provided implementations of the interface which receives behavior configurations (sets of events with associated actions). Moreover, more complex processors can be created by writing specific implementations. The following is an excerpt of the interface’s code:

```java
public void sendInformationToken(String receiverName, String tokenName, String product, int quantity, String originalSender);

public void sendValueToken(String receiverName, String product, int quantity, String owner);

public void sendMoneyToken(String receiverName, String subtype, int quantity, String owner);

public void weakUpForPeriodicTimeEvent(int time);

public void weakUpForSpecificTimeEvent(int time);
```

**Grocery Store Experimentation**

Given the blueprint of the Grocery Store, we defined the behaviors of the corresponding processors implementing IBehavior. To do so, we defined 13 Java classes that implemented the IBehavior interface. For each processor, we identified the relevant specific and periodic time events along with the processors
that relate to it. We present an example of the implementation of one of the IBehavior methods for the Warehouse Manager for a periodic time event.

```java
public void weakUpForPeriodicTimeEvent(int time) throws Exception {
    switch (time) {
    case 5:{sendInformationToken("Store", "Order", "Perishable goods", 30, me.name); sendInformationToken("Store", "Order", "Non-perishable goods", 30, me.name)} break; }
}
```

To evidence the way in which the blueprint serves as an input for experimentation, we used the grocery store blueprint execution to test how the cash in the store is affected by changes in client's demands and orders to suppliers. We expected to visualize whether cash increased or decreased and the rate in which it did.

We ran an initial simulation of the blueprint, and used the cash accumulation traces to graph the overall behavior over 45 months. We then changed the parameters for three experiments, in the behavior of the Store and Delivery Clients, and in the Warehouse Manager respectively. The parameters that varied are presented as follows:

- **Experiment 1** Double store client order rate.
- **Experiment 2** Half delivery client order rate.
- **Experiment 3** Increased nonperishable order rate.

Figure 5 presents the results after executing the three experiments. As it is evidenced, decreased orders and increased orders to suppliers reduce cash, although a decrease in orders leads to less cash levels than extra orders to suppliers. Doubling client’s orders while increasing the cash levels do not double them in comparison to the initial scenario. It is important to note that said results provide a valuable insight for the study of the business and while they do not necessarily serve as forecasts, they allow to understand the cause effect relations embedded in the exchange network.

![Cash Analysis](image)

**Figure 5 Cash Analysis**

**Conclusion**

This paper proposed an executable business model blueprint to represent the structure and the dynamics of a business model. This blueprint represents the unification of several existing approaches for the description of business models, and it is based on 8 building blocks which were derived from previous works. This blueprint also provides the means to describe the key dynamic aspects of a business model. In particular, it serves to describe the behavior of agents in terms of events related to flows and actions.

This paper showed that the blueprint provides the means to generate a better and more profound understanding of a business model. For example, it is possible to observe how the business in general, and each agent in particular, reacts to internal changes and to external forces. This makes the blueprint an effective tool to experiment and study the behavior of a business model in different scenarios. Although it cannot be considered a forecasting tool, it provides valuable information that can be of help in complex situations such as when new business models are being designed, or when existing models are being transformed.

The paper also presented a small analysis tool based on the business model blueprint. We are currently working on more advanced analysis methods and platforms. In particular, we are working on the definition
and implementation of a business model laboratory for start-ups, where business models will be described using the blueprint and then subject to experimentation. We expect this laboratory to be of use for emerging enterprises by revealing hidden relations between elements in the business model and the surrounding environment, showcasing potential problems and opportunities, and ultimately reducing their failure rate.

A final remark about the presented work is connected to the complexity of business models and the additional complexity introduced by the dynamic aspects. As in any other modeling approach, a trade-off exists between the complexity of the models, their usefulness, and the construction costs. However, we believe that this problem is not meant to be solved in the model itself; instead, it is necessary to develop a methodology for using the blueprint that will ensure that sensitive decisions are made with respect to the level of detail to include in the models. This is akin to problems faced when using Osterwalder’s canvas (e.g., Which activities are important enough to be included in the canvas?) or a language like BPMN (Is this action a task, a subprocess or a process?).

REFERENCES


