Visualizing Business Model Evolution with the Business Model Canvas: Concept and Tool

Boris Fritscher
Faculty of Business and Economics
University of Lausanne
Lausanne, Switzerland
boris.fritscher@unil.ch

Yves Pigneur
Faculty of Business and Economics
University of Lausanne
Lausanne, Switzerland
yves.pigneur@unil.ch

Abstract— **The Business Model Canvas (BMC) assists in the design of companies' business models. As strategies evolve so too does the business model. Unfortunately, each BMC is a standalone representation. Thus, there is a need to be able to describe transformation from one version of a business model to the next as well as to visualize these operations. To address this issue, and to contribute to computer-assisted business model design, we propose a set of design principles for business model evolution. We also demonstrate a tool that can assist in the creation and navigation of business model versions in a visual and user-friendly way.**

Keywords—Business Model Canvas; Business Model Evolution; CABMD

I. INTRODUCTION

The Business Model Canvas (BMC) [1] is a visual modeling method that is used to capture the business model of a company. It is defined by nine building blocks. The method involves adding sticky notes to each of these building blocks; these note represent the elements involved in the business model. A completed BMC will highlight the key elements of a business model at a fixed point in time, as chosen by its creator. The BMC has achieved widespread adoption [2]: not only is it used to model the current state of companies' business models, but also any future business model innovation. When a company undergoes changes, these need to be reflected in its business model. Planning for new strategies also generates new business models of possible future states. Currently, every canvas is standalone. Thus, for each canvas a specific time context is chosen: past, present, or a possible future. Even if changes to a business model can be represented by a new BMC, these two canvases are not linked together and there is no indication as to how to compare them. Identifying the changes that have occurred between the two states is difficult. It is necessary to review each element for both canvases, comparing them one by one to see if they have been added, changed or removed. A more efficient method of highlighting changes between two canvases is needed, one that creates a visual roadmap of a business model's evolution. Only then, will companies be able to better understand business model transformation.

A. Business Model Canvas

In this study, we chose to focus on the BMC because of the problems raised through its broad adoption, and because its visual and simple-to-use structure aligns with design thinking and managing as designing [3-4]. Proficient users are able to do more than just add elements to a model's building blocks; they can also consider element interactions and multiple models [5]. For more advanced users, however, it is necessary to develop a method for dealing with changes to a business model.

B. Business model mechanics

An expert BMC user can highlight key business model mechanics and interactions between elements by drawing arrows to connect these elements. Business model evolution affects elements, so any changes to these elements will also have an impact on mechanics. When new elements are added or removed, some mechanics may become more important or less important.

By taking mechanics into consideration during the design of a business model, new ideas can be generated, leading to the exploration of new business models. However, a solution is still required to assist in the handling of multiple models and their transformation.

C. Business model evolution

Our particular interest is in the transformation that occurs between two discrete evolutionary steps of a business model. In reality, changes occur in a fluid way, on an ongoing basis. We consider the issue of grouping changes into sets of mutations to be outside the scope of this paper. The choice of decomposition into distinct steps is carried out by the designer, who uses his own methods.

A business model's elements will change at some point. At the time of change, the overall business model will be transformed so that it is distinguishably different from its previous version. Each small change can be seen as a mutation of an element, which is either an addition, a removal or a modification. A set of mutations is the difference between two versions of a business model. This provides us with the means to consider, for example, the history of a business model as an evolution through different states or as ticks on a timeline. Changes may occur as the result of internal research and development, or through the continuous improvement of tasks, and reactions to changes in the business model's environment. While this sequential vision works for tracking past and current states of a business model, a more flexible approach is needed if we wish to explore future states, of which there are more

than one. Future business models represent possible versions. These have been imagined in response to future changes that have resulted from internal innovation or a reaction to environment changes and can be discovered by, for example, simulation or scenario-based prediction tools. An evolution graph looks rather like a tree, with branches representing the possible future directions of the business model. One of these branches might become the next current state. Having a better description of the evolution that is required to reach such a possible future state opens up the possibility of evaluating the necessary transitional steps, thereby allowing us to identify the most suitable target for the next evolution step.

D. Computer-aided business model design (CABMD)

Handling all these mutation steps on paper has its limitations. Marking up changes to previous states requires elements to be redrawn. What's more, hiding deleted elements is difficult. A CABMD tool would be much better suited to the handling of such dynamic design. Context changes require a different selection of visible elements to be displayed. By enabling the previous model to be visible in the background, it is easier to identify which mutations are needed to transform the current business model into a new one. As a result, only the end result, or the transformation can be displayed. Moreover, being able to switch the view in an instant is in the design spirit of prototyping and helps in the iteration of different mutations until the most suitable model has been identified.

Features derived from CAD tools in other domains, such as architecture, process design, and programming, can be used as an inspiration for CABMD. However, special care has to be taken to design accessible tools that do not require special training or a technical background. The main user group consists of business people, who tend to be less technically aware than traditional engineers [2]. Likewise, the tool should have as much built-in flexibility as possible, offering advanced designers constraint validation of the model, whilst still offering some guidance to novice users.

E. Research questions

The focus of this research can be summarized by the following questions:

- What would a formalized set of concepts for mutating from one business model to another look like?

- What would a visualization of a Business Model Evolution with discrete steps look like?

- What design principles does a CABMD tool need to fulfill in order to support Business Model Evolution visualization and manipulation?

In terms of structure, this paper follows design science research guidelines [6]. First, we present the justificatory knowledge, followed by our methodology. We then discuss the artifact (i.e., the use case and our concepts), before going on to describe a prototype that implements the artifact, followed by an evaluation of that prototype. Finally, we discuss the implications of the methods used and conclude with a look forward at further research on CABMD.

II. JUSTIFICATORY KNOWLEDGE

In highly uncertain, complex and fast-moving environments, strategies are about insight, and rapid experimentation [7]. This impacts on the need for business model evolution. Indeed, the adaptability of a company's business model is key to its ability to survive [8]. However, as yet, no visual or technical methods to transcribe these changes on to a business model have been put forward. From a technical perspective, evolution can be interpreted as a series of business model states that are versioned. Ideas and technique for versioning are not new. They have already been covered extensively in many technical disciplines. However, none of them fit perfectly to the business model evolution tracking needs we have identified. With our focus on CAD we highlight only briefly two examples from an IT focus, there exist however also other fields such as architecture and other design activities.

In fields such as database modeling, schema evolution is well known in relational and object-oriented systems [9]. Solutions exist, but require a technical expertise or are not easily adapted to high-level and managerial-level business illustrations.

Versioning of work was first introduced in development environments and has proved successful in handling a variety of change tracking that goes far beyond the initial source files [10]. At its core, versioning serves to guarantee the integrity of the changes, giving the business model an immutable history. However, whilst this works for documenting the history of an evolution, it does not support creative prototyping, where a flexible and dynamic back-and-forth way of idea exploration is required.

III. METHODOLOGY

This study followed a design science research approach [6]. We identified a need for a structured method to visualize business model evolution using a BMC. To address this need, we elaborated a series of concepts, which were then instantiated with different prototypes in order to iteratively improve and evaluate them.

The iterations to produce the design solution were carried out under a set of constraints because our solution has to be suitable for use with a BMC. Thus, it has to follow the principles of ease of use and of visual design [2] to guarantee that it will be accessible to our target audience of business users, who do not have special CAD tool training.

To get a better grasp of the concept of business model evolution modeling, we started by gathering insights. We also built a use case, which served as the foundation for creating the first visual prototype. The use case was built by carrying out desk research about a company that had been identified as having multiple evolution steps. All the available data was aggregated to build discreet identifiable evolution steps, which were then described textually. In a second stage of the process, key business model components were extracted from each step, giving us a series of BMCs, which could be linked together. This visual prototype was then iteratively evaluated and refined. The key transformation concepts were extracted from it and an insight was gained into their design principles.

The visual prototypes were built using the core tenant by which paper-based design is combined with the possibilities offered by computer-assisted software. We explored the idea of layers for each step of business model evolution by evaluating different prototypes: first with paper experiments, then with illustrations created using a digital painting tool and finally with a custom software tool. Each prototype gave additional insight, enabling us to solve the issue of visually displaying these steps and the transitions that take place during business model evolution.

IV. CONCEPTS FOR BUSINESS MODEL EVOLUTION

In this section we describe concepts of business model evolution, starting with the basic operation to transform one business model state into its next version. We then show how the concept of layers can help with visualizing these operations, extending this transition to multiple iterations. We can then examine how this affects the representation of different layers.

A. Mutations classified

To describe the transformation from one business model version (layer) to the next, it is necessary to look at three mutations: the add operation, the change operation to transform existing elements, and the remove operation, which complements the add operation.

These actions are very similar to those proposed in the Blue Ocean Strategy 4 Actions Framework: Create, Eliminate, Raise and Reduce [11]. In this study, however, we describe how mutations can be used to define the transformation of one business model (previous, layer 1) into a new one (layer 2).

1) Add
Adds a new element to one of the nine building blocks of the BMC. Something new that the previous business model did not utilize.

2) Remove
Removes an existing element from one of the nine building blocks of the BMC, completely discarding something that was used before.

3) Change
Any element that behaves differently or has one of its properties transformed. Instead of considering two distinct cases, such as raise and reduce from the 4 Actions Framework, we used a generic change operation. On a basic level, change could engage the remove and add operations; however, this would not allow existing mechanics to be involved in the changed element without the removal and recreation of both the mechanics and connected elements. The change mutation in operations allows us to trace the origins of the elements and the state where they were introduced to the business model. The change mutation also shows how the business model evolved into its current state.

For some changes, it is helpful to indicate if it was a raise or reduction. This can be specified as an additional indicator, without noting whether it is a positive or negative increase, because this depends on the context.

4) Other mutations

More complicated concepts, such as the notion of splitting one element into multiple elements, or the merging of multiple elements into one, could also be considered. On the other hand these operations can be handled with a combination of remove, add and change operations; for example, by deleting all merged elements and adding a new one, or by deleting all except one and then changing it. The caveat is, of course, that it may lose some evolution information. Although we chose to introduce the change event for these reasons, in a situation where there is merge and split, any added complexity would outweigh the benefits provided.

B. Visual representation with layers

In order to group mutations into business model states we used the concept of layers inspired by 2D animation techniques. Here, layers are used as a stack of transparent sheets, where each one represents a still image frame of the animation. A new frame is a new layer, which is used to draw over previous areas to illustrate movement through changes. Applied to business model evolution, the use of this technique is similar to that of drawing on tracing paper. The mutations are applied to the new layer, thus affecting elements that are visible on the lower layers. Here, the redraw is not based on the location of the painting canvas, but on the process of adding, removing or changing business model elements in the nine building blocks. The operations of the current layer are shown normally; the previous layer is shown as faded.

C. Multiple evolution steps

A business model evolution that involves more than one transformation adds some complexity, which we address here.

1) Tree of Business Model Evolution
Stacking more than one layer shows a chain of evolution for a business model, as well as a series of transformations. Furthermore, it is also possible to avoid stacking all the layers directly on top of each other by sharing some layers and then branching off into another direction, much as branches grow on a tree.

2) Import
However, if there are different branches and one element is to be shared from one branch to another without adding it to the common parent branch (layer), the change operation does not fully cover this use case. Therefore, we have used the concept of import to highlight an element that is used from another layer when there is no common element in a shared parent layer. This is a special version of the change element. It was also used in our use case in which two business models were developed in parallel, sharing some elements. This concept is necessary because of dependency problems when there are more than two layers. Other consistency rules which did not require a new concept are covered in the design principles section.

3) Multiple layers aggregation rules
Visually, a good way to understand business model evolution is to compare the difference between two consecutive layers. When there are multiple layers, making such a comparison can be difficult because of visual crowding, especially if there are a large number of change and delete operations across the layers. One way to compare a layer with

the previous layer is to aggregate everything except the two layers that need to be compared. This aggregation creates a simplified previous layer and helps to declutter the visual design. It also means that if an element is removed in the previous layer, it will not show up in the current layer. When changes have been made, only the final changed element is displayed; the most recent change overriding the older ones. All the elements of the previous layers have the same fade level; they do not fade more and more, as if they were layers of paper stacked one on top of the other.

V. DESIGN PRINCIPLES FOR COMPUTER AIDED BUSINESS MODEL DESIGN

Supporting a design methodology with CAD tools has several shared advantages: guaranteed consistency of the model, easier visual navigation with multiple representations of the same data, and support for explorative work methods with the ability to move elements around at no cost. In addition, a number of principal benefits can be identified for each concept; these can be implemented by a CABMD tool to support business model evolution.

A. Transformation principles

P1: Consistency of the supported model

The tool should allow for flexibility while still guaranteeing automatically the consistency of the model it supports. Applied to the business model concept we have the following rules:

- A series of changes has an add mutation as the initial parent element;

- A link always connects two existing elements;

- An import is a change element that has a parent element to which it applies a change. Which is not in any parent layer of the layer the import is in.

Consistency can be guaranteed in different ways, as shown.

- By blocking any operation that would lead to an inconsistent state. For example, to guarantee consistency, one should not authorize the removal of elements involved in a link before the link itself is removed.

- Draw to the attention of the designer the problem to be resolved; the designer can then make a decision. For example, he could choose to cancel the requested operation or decide to remove the element and any relevant links.

- If the solution is obvious, any corrections that are needed to make the model consistent can be applied automatically.

B. Visual principles

P2a: Visual representation and simple navigation

Any interaction has to be a simple and visual representation of the business model, and must be understandable at a glance. A CAD tool needs to be easily editable and support visual color coding of the elements [2].

P2b: Visual mechanics

Visual mechanics illustrate the influence of elements on each other in the BMC. Semantically, it can have different meanings, depending on the story being represented. Therefore, the implementation of visual mechanics has to be open and allow for a drawing-based system, whilst still enforcing the connection between the start and end elements of the link. Links are limited to connecting elements in the nine building blocks of the same BMC; however, they can span elements of different layers.

P2c: Layers as business model states

In terms of replication, digital layers help to fade previous models in a way that is smarter than using tracing paper. Instead of having the lower layers fade to nothing, each layer can be computed so that it is visually "readable", with the right amount of transparency. In addition, it is relatively inexpensive to use a digital format to improve visuals by displaying the same information multiple times and in different ways.

In Business Model evolution each layer represents a new business model through the aggregation of the previous layers. Instead of putting everything on top of each other, each aggregation (state) can be displayed in a sequential way, as a chain of business models, or can be spaced along a timeline.

C. Consistency during design dynamics with multiple layers

In this section, we describe the rules that govern the movement of elements whilst the modeler is experimenting with the design, moving elements between layers before reaching a final solution.

P3: Mutation rules on business model elements

Elements that are new (add elements) can be moved to all other element types (change, import, remove), or they can inherit their position from the element they modify. Moving an element's position inside the same building block only has a visual impact. Switching building blocks inside the same BMC changes its element type. If an element is moved into a layer that represents a previous state of the current business model, everything is fine. If the element is moved to a later business model state, which already includes a change element there may be a problem; this depends on the moved element. The elements that are added and changed will occupy the same space. Thus, it is necessary to decide how to merge them or whether or not the move is invalid.

Moving an add element out of the business model inheritance tree or onto another business model branch, might leave change elements that do not have a visible add parent. The first change element needs to be transformed into an import. Alternatively, the move will have to be cancelled and the user alerted as to existence of a dependency problem.

VI. USE CASE: VALVE CORPORATION

We developed a use case based on an American company named Valve Corporation. They started out as a classic video game development studio and transformed themselves into the leading actor of digital video game distribution (Steam[1]). From

---

1 http://en.wikipedia.org/wiki/Steam_(software)#History

a business model point of view their case is interesting because we could identify seven evolution steps within twelve years and spanning two business models: game design and distribution platform. The latter emerged from the first one, but is more than just an extension. Identifying the states and their transitions was in itself a challenge. Part of the final prototype tool can be seen in figure 2.

A. Example of a transition

In 2011, Valve Corporation started experimenting on applying the free-to-play business model concept to one of their more successful multiplayer games, "Team Fortress 2". This meant that they no longer sold the game, but offered it for free to anyone to download and play. The lost revenue from game sales was replaced by in-game micro transaction sales of add-on items for the in-game avatars: new decorative hats, glasses, and weapons. Additionally, they approached independent developers, offering them the possibility of producing new content for the game through a revenue-sharing model.

Key mutations for this transition were two new value propositions: "free game" and "in-game items market". In addition, there were two changes in the customer segments targeted by these new offerings: "gamers" which expanded to include more casual players and "independent developers" which increased in terms of numbers and professionalism, as the developers were attracted by the new revenue possibilities.

VII. INSTANTIATION

The built prototype is a web application (see figure 2). It provides a workspace on which the BMC and its layers can be added. Each layer aggregation is shown on a sequential timeline. Navigation between the states can be achieved by zooming into the workspace and panning around.

The following features support the design principles described in the previous section.

1) Consistency

The user interface limits element creation on a layer to given operations. On any BMC, elements can be added to any position of any building block. Changed or deleted, however require clicking on an added element from a previous layer.

2) Visual principles

a) Visual representation and simple navigation

Each element has a set of visual properties that dictate its appearance, such as name or picture, and a set of colors. We chose to use a multicolored tagging system [2], which is recommended for this design. Elements have a fixed size, but can be freely positioned.

Since the visual appearance of an element is computed in real time it is easy to provide options that customize the look of any displayed information. A number of visual choices can be turned on or off: links between BMCs, the mechanics in a BMC, and the previous layer can all be shown. Likewise, customization can be used to show only the new mutations or the final model without any fading. The space between business model states can also be changed to influence its

position in the tree, or in order to simulate a timeline over a purely sequential layout.

b) Visual mechanics

Links between elements are created by dragging and dropping one element on top of a second element. The visual appearance of a link can also be changed by specifying its stroke width and dash style in addition to its color. The start and end points of links are confined within the bounding box of their respective elements. A link can have as many intermediary nodes as necessary to draw a custom BM mechanic path between its start and end elements. This allows the BM to express an indirect route, which visualizes the usage of other elements (limited to only visual information).

c) Layers as business model states

A BM can create a new child business model which inherits its elements; here, links are faded. The child model takes the form of a layer that is positioned on top. New transformations are recorded on this layer.

3) Multiple layers

A business model can have as many children as necessary; each child is a new branch in the tree. A child business model can, in turn, give rise to another child business model, thus creating a chain, with cumulative results for each branch.

An existing element is inherited from its parent chain (either from a direct parent or a grandparent). Such an element cannot be moved directly; instead, it can be marked as removed or changed according to the rules of mutation. A change can optionally specify whether it is an increase or decrease on its impact indicator. A changed element inherits the position of its parent element, but can have its own name, and color. An import is similar to a change, except for the fact that its business model is not in the same business model chain. A deleted element will no longer be visible in any child layers.

All links that involve the element marked as deleted will also be hidden for child business models in this branch. Links can be drawn between new and changed elements, as well as previous ones.

VIII. EVALUATION

Each designed prototype during this research was evaluated before its iteration to produce the next one.

First, we used tracing paper to create a paper prototype visualization of the layered business model concept in the use case. We then asked a small group of users who were already familiar with the business model whether they could understand the different visual cues offered by the layers. This confirmed that the visualization in layers is helpful, but showed that the use of paper had its limitations: when stacking a series of physical pieces of paper on top of each other, the underlying elements become faded. A digital prototype can remediate this by offering more selective filtering. Thus, we went on to iterate with the help of a painting application. This allowed us to test different combinations of transparency, color and layout by gathering readability feedback from test users. The display of layers visually helped in the gathering concept and in the design of the business model. These visual layers were then

applied to the creation of a software prototype. Although the static painting used a digital format, it made sense to have the additional functionalities offered by a CAD tool in order to selectively show or hide a layer's elements based on a given perspective. The software prototype was first tested to validate if it was in fact possible to design the whole use case with a paper-based prototype. The use case is a complicated business model evolution that takes place over seven steps and involves two business models evolving in parallel. Testing was extended to visualizing student projects in order to validate interactions with the tool. The final prototype was tested by students, who used it in their business model course design project.

A. Experience setup

Nine groups of students from a master's program in IS produced a business model proposition for a startup company. They created multiple iterations of this business model in order to, participate in the trial. They were asked to recreate their business model iteration with the business model evolution concept, using the prototype provided. Before allowing the participants to use the tool on their own, a short demonstration was given in order to explain the key evolution concept and related user interface functionalities. Some known limitations of the prototype were also highlighted. The students had already separated their business model into discrete steps for their project. Their task consisted in entering their first version of their business model and then extracting and adding the differences between the first and next business model onto a new layer. This process was repeated for each iteration. Being already familiar with their business model, they could focus entirely on the visualization and mutation operation aspects of the method without having to spend time thinking about identifying the iterations.

B. Observations

The students were observed during their usage of the tool. They could ask questions about the theory and interface. Once they completed the task, the students had to fill out a survey, which included open questions on topics of usability and the modeling process. They also had to answer four questions from a seven point Likert scale about perceived usefulness.

1) Usability

The students had no problem in adapting to the tool. The short presentation of ten minutes given at the beginning was enough for them to be able to understand the concept and any interactions. All groups finished their business model evolution input in less than an hour. Most of the time was spent discussing aspects of business model evolution that emerged from the modeling, rather than inputting the model into the tool, which itself was fast. Whilst there was some discussion about merging, the add, delete and change operations were all that was necessary to represent all business model transitions.

2) Modeling process

We then asked if the business model evolution exercise helped the groups identify missed opportunities or incorrect transitions from one state to another. Using this information, we identified two categories with interesting correlations. Those groups that reported no additional opportunities or errors from their work, were also the groups who were awarded high evaluation marks for their project by their teacher. Likewise, the groups that received a lower evaluation of their project reported that they did identify ways in which the tool could have helped them with the transitions.

Other feedback related to the enabling of a better global picture using BM evolution visualization:

G05 "Was able to better identify connected elements between iterations"
G10 "See where it came from"
G08 "Gives a good general overview on the evolution of our idea"
G09 "Helped us to view the changes and at which time/iteration they occurred"
G03 "It allowed me to analyze the reasons behind the evolution of our business model. It's also helpful for the storytelling."

3) Perceived usefulness

All groups perceived that it was useful to track business model history using the tool and reported that they would have used it for their project. This is the task which is most similar to the one they were asked to do for the evaluation. They also said they were very likely to use it for their next project.

However, perceived usefulness for prototyping alternative business models scored only average. One explanation is that this feature requires advanced knowledge of the methodology and more practice in thinking about alternatives. This is a concept that they only saw briefly during their course; they were not able to put it into practice in their group project. An example that illustrates the difference in modeling capabilities of novices and experts was the way in which one group struggled to model their project history. They tried to explain it in a linear fashion, using three business model evolution steps. On the third layer, they wanted to add the same elements that they had removed on the second one. The tool's coherence validation makes this operation difficult, indicating a possible flaw in the design. An expert BMC user suggested that the second and third layers could be modeled as parallel evolutions, rather like two branches that emerge from the first layer but which explore different directions. This solved the problem and produced a design that is more accurate than that offered by the linear solution.

IX. DISCUSSION

In this section, we review the different kind of usages enabled by business model evolution and described using layers. We also discuss the impact this has on mechanics and the potential of the layers concept for business model design.

A. BM Evolution Usages

We discuss four ways that layers are used in a business model.

1) History / Evolution of a business model

Knowing the origins of elements and the influences behind any changes helps to better understand the constraints of the current business model. Moreover, it offers a way to identify an evolution pattern. In our use case, we observed a cyclical back-and-forth movement between the resources and the value

proposition. Resources are used to offer a new value proposition (game distribution platform), which in turn create a new resource (sales knowledge). This resource can then be used to evolve the business model offering with new services (micro transactions).

2) Possible alternative future scenarios

Business model evolution should not be seen only as a linear sequence. Each state can have multiple children, which all have different changes branching out of them - much like a tree - allowing them to adapt their business model to different scenarios.

3) Transformation inside a business model

The identification of transformation and visualization with sequential layers can also be used to illustrate the workings of a business model. Instead of transforming from one version into another, one business model is separated into stages (layers). Each stage is not a complete business model, but a component or a timeframe in the execution of the business model. Illustrating the business model in multiple stages help to better understand its mechanics. The more dynamic nature of flows such as financials can be made visible. For example, it can help to visualize the difference between manufacturing for stock or to order; this was the basic business model pattern for HP and DELL respectively.

In cases where computers are produced and then stored in the retail channel, costs have already been occurred, as can be seen in figure 1. Revenue is only generated in the second stage, when sales occur. Cash flow is negative, as shown by the left to right arrow. In the case of manufacturing to order, however, the revenue is collected first, with costs occurring in the second stage. Cash flow is positive, as shown by the right to left arrow.
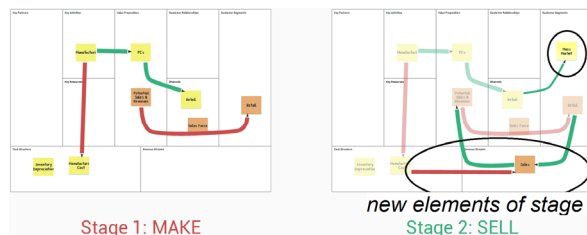

Fig. 1. Example of PC manufacturing

4) Transformation between business models

The mutation concept can be used to compare two separate business models without them having to be versions of each other. Having a set of defined concepts for a business model and a set of concepts that specifies how to transform from one state to another gives a common language with which to compare elements. The difficulty lies when making comparisons between the elements. A great deal of visual complexity can also be quickly generated when a lot of elements change. If everything is different, everything will be an addition or a delete. What's more, text-based elements with keywords can be different, but still mean something similar. The inverse is also true; the same keyword can have a different meaning depending on the context in which it is interpreted.

B. BM mechanics when designing with layers

By highlighting a business models' mechanics using links that flow between key elements, important visual cues can be given, thus making the business model more understandable. To give a more structured meaning to these links, we classified the interaction between the different possible states of mutation for the elements involved. An element can either be unmodified from a previous state, it can be changed or it can be new. Removed elements cannot be linked with a new mechanic. It is important to consider previous elements because the links can have associative meanings that connect multiple elements together into a chain. The following series of combination possibilities can be identified as follows:

1) From a previous element to a previous element

A new link between existing elements cannot be created in the current business model layer. At least one element has to be different from the previous element.

2) From a previous element to a changed element

A change in the destination element makes it more important for business model mechanics, so a new link is created.

3) From a previous element to new element

An existing element targets a new element without having changed. A new value proposition is consumed by an existing customer segment. An existing resource is used to provide a new value proposition.

4) From a changed or new element to a previous element

A changed or new element has a reinforcing influence on an existing element, but not enough to warrant a change in the element itself. However, an element further up in the link chain can be changed. In our use case, new resource modifications influence the multiplayer gaming value proposition.

5) From a changed element to another changed element

A changed element leads to another changed element; they might have already been linked, but the link becomes stronger. In our use case, a change in automatic patching and new game engine influences a change in customer segments to a wider pool.

6) From a changed element to a new element

When an existing element changes, this offers an opportunity for a new connected element. In our use case, a larger number of independent game developers leads to the production of new in-game items.

7) From a new element to a changed element

A new element can have a strong influence on another element to the point of changing it. In our use case, the new distribution platform changes game patching customer support dramatically through the use of automation.

8) From a new element to a new element

When a new set of elements becomes connected, new opportunities can result.

The direction of the link is not always obvious; indeed, depending on the perspective taken, it can be inverted. A link can be read either as something being produced or as something consumed. It is often the case that the illustration of business model mechanics does not follow a strict rule of adopting one vision continuously for the same BMC.

For example, in the use case, a link is shown between the customer segment, the increase in independent developers, and the production of a new value proposition (in-game items). Alternatively, this could be viewed as a new value proposition, which offers independent developers a new way to make more money, and which changes (increases) this customer segment. In the use case, the direction was chosen to show the flow of the produced items that reach the end customer through the game. The opposite direction could have been chosen to illustrate the financial flow. Therefore, strict rules that govern the complete link chain should be avoided during the design phase. Sometimes the directions are also chosen to best fit the explanatory story of the business model.

C. Other ways of using layers

In this paper we used the concept of layers to group visual elements and to filter parts when stacked. However, there are other ways in which layers can be useful to business modeling. A layer could, for example, add functionality or new attributes to the involved elements (like a trait). In mapping software different layers allow us to see different aspects of the same object: for example, satellite maps, terrain representation, and 3D buildings. Transposed to business models, there could be new layers for indicating the validation status of an element. Layers could also be used to add social consideration. A calculation layer could show cost and revenue information for involved elements.

X. CONCLUSION

In order to address the issue of managing business model evolution, we started by building a use case. Using a case-based approach instead of abstract thinking about the generic problem was crucial to our iteration on the visual concept. It helped us focus on a real situation with concrete problems. The design principles and concept extracted from the use case were then taken to build a new prototype, which was evaluated with different cases to test the concepts. The resulting actions of adding, removing, changing and importing elements when moving from one business model to a new one helps to formalize BM evolution. Building the prototype and successfully testing it demonstrated the feasibility of the concept. Beyond this, the design principles and sample implementation helped in advancing the specification of the domain of computer-aided business model design.

Using the same tool, we managed to support novice users, encouraging them to perform better, while also supporting expert users with more advanced features. However, we also observed that there still is a risk of misusing advanced features. Thus, it is necessary to engage in further research into teaching best practice to users and to identify integrity rules that can be checked.

Further research should extend the evaluation of such tools. In particular, work could be carried out to look into how the tool can help to identify the distinct evolution steps of a business model. Future research could also evaluate how the concept and tool perform with regard to the idea generation task. Furthermore, applying the concept to other strategy methods, would allow generalization and help in strengthening the emerging domain of computer-aided design for strategy.

REFERENCES

[1] Osterwalder, Alexander, and Yves Pigneur. Business model generation: a handbook for visionaries, game changers, and challengers. John Wiley & Sons, 2010.

[2] Fritscher, Boris and Yves Pigneur. Computer Aided Business Model Design: Analysis of Key Features Adopted by Users, Proceedings of the 47th Annual Hawaii International Conference on System Sciences, Computer Society Press, 2014

[3] Boland, Richard, and Fred Collopy, eds. Managing as designing. Stanford University Press, 2004.

[4] Martin, Roger L. The design of business: why design thinking is the next competitive advantage. Harvard Business Press, 2009.

[5] Fritscher, Boris and Yves Pigneur. From Business Model Ontology to Business Model Canvas, (submitted), 2014

[6] Gregor, Shirley, and Alan R. Hevner. "Positioning And Presenting Design Science Research For Maximum Impact." MIS Quarterly 37, no. 2 (2013).

[7] McGrath, Rita Gunther. "Business models: A discovery driven approach." Long range planning 43, no. 2 (2010): 247-261.

[8] Teece, David J. "Business models, business strategy and innovation." Long range planning 43, no. 2 (2010): 172-194.

[9] Nguyen, Gia Toan, and Dominique Rieu. "Schema evolution in object-oriented database systems." Data & Knowledge Engineering 4, no. 1 (1989): 43-67.

[10] Tichy, Walter F. "RCS—a system for version control." Software: Practice and Experience 15, no. 7 (1985): 637-654.

[11] Kim, W. Chan, and Renée Mauborgne. "Charting your company's future." Harvard Business Review 80, no. 6 (2002): 76-85.
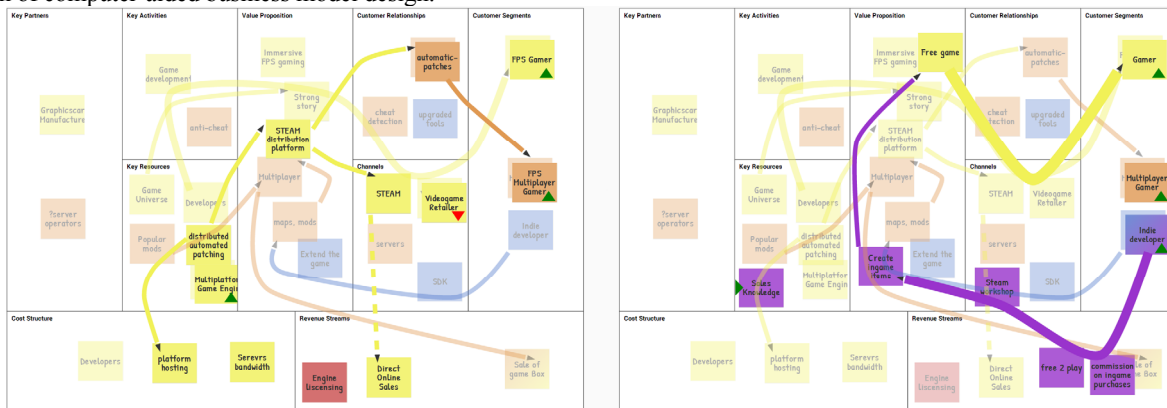
Fig. 2.   Screenshot of parts of the prototype showing the use case modeled: a video game company that evolved its paid game offering into a free-to-play game with micro-transactions. Operations: add free game, add create in-game items, change indie developer, change gamer.