A Method to Support the Alignment of Business Models and Goal Models

Pierre Halleux[1], Ludovic Mathieu[1], and Birger Andersson[2]

[1] Faculty of Namur, Department of Computer Science,
rue Grandgagnage 21, 5000 Namur, Belgium
{phalleux, lmathieu}@info.fundp.ac.be

[2] Royal Institute of Technology, Dept. of Computer and Systems Sciences,
SE-164 40 Kista, Sweden
ba@dsv.su.se

Abstract. This paper addresses one part of business and IT-alignment by proposing a method to align goal models and business models. The method takes as input a goal model and a business model, and outputs a business model that is aligned with the explicit goals of a business actor. The method builds on previous work with the same approach but extends that work in at least two ways: the syntax of some method constituents is altered and a way to combine them is introduced. The result is an improved method that better support a modeller when designing business models based on goal models.

1 Introduction

Generally, the raison-d'être of ICT in an organization is to support the organization's business goals and this is often materialized as IT-systems for support of operational processes. The goals should therefore be made so explicit that supporting IT-systems can be aligned with them. A problem then is how to formulate business goals so that the alignment can be made. One solution approach is to utilize models that focus on different aspects of an organization and its collaborations. Addressing the problem then amounts to aligning a chain of models.

A common view is that goal models are used in the earliest phases of business and information systems design, where they help in clarifying interests, intentions, and strategies of different stakeholders answering to the "why" of the business. Business models give a high level view of the activities taking place in and between organizations by identifying agents, resources and the exchange of resources between the agents. So, a business model focuses on the "what" of a business. Process models focus on the "how" of a business, as they deal with operational and procedural aspects of business communication, including control flow, data flow and message passing.

The purpose of this paper is to present a method for addressing one part of the problem of aligning the IT-resources with the goals of an organization — the alignment of goal models and business models. The method approach is to use templates for formulating goals and apply rules for business model transformations. The method builds on work presented in [1]. As that work outlined the method on a high level a research question was how to make it more formalised. This work extends previous work in several ways, most notably in the amount of formalism used. In this paper we illustrate the extended method in a case application making use of two well known modelling techniques; for goal modeling we use i* [2] and for business modelling we use e^{3}value [3].

The amount of research and literature pointing out the importance of business and IT alignment is vast. Notable examples of approaches for alignment through model use can be found in [4], [5], and [6].

This paper is structured as follows: the method is presented in section 2. An illustration of the method by its application on a small case is in section 3. Finally, the concluding section 4 contains a discussion of the results and directions for future research.

2 A Method for Goal and Business Model Alignment

A common problem in goal modelling is that goals are difficult to formulate, i.e., the formulations of goals often become loose, highly abstract and unfocused. In [7], the authors argue that goal models become unfocused because goals range from the value propositions of an enterprise to general goals of economic sustainability. However, largely all means in goal models (a means is an action carried out to attain a goal) relate to the acquisition, production, maintenance, or provisioning of economic resources. As mentioned in section 1, business models describe the use and exchange of resources that are of economic value for agents participating in collaborations. We exploit this relation between means and business model notions when formulating the following method for goal model and business model alignment.

2.1 Method Overview

The method, originally introduced in [1] but here substantially extended, takes as input an as-is business model and a to-be goal model and produces a new to-be business model conforming to the goal model.

The method has two main steps, where the first concerns goal modeling and the second concerns business modeling. In the first step, it is the responsibility of a goal modeler to construct a goal model using business model notions; in particular the means are formulated according to a template structure (see Sect. 2.2). In the second step, it is the responsibility of a business modeler to make use of the means supplied by the goal modeler by applying transformation rules to a business model. If the business modeller do not have required information to apply a rule, then this information must be elicited in order to continue. The method can be outlined as:

1. The goal modeler constructs a goal model using the means templates.

2. For each means template the business modeler:
   (a) complements the means by filling in the required and optional parts when needed.
   (b) applies the relevant transformation rule.

For each means template, there will be exactly one transformation rule telling how means of this template will influence the to-be business model. The means templates can be categorized into three main groups based on their effects on the to-be model: templates leading to the introduction of new business model components, templates leading to the deletion of certain business model components, and templates requiring changes at the process level (see Sect. 1). While the first two groups have a visible effect on the to-be business model, the effects of the means in the third group is not visible in this model but will only have impact on a process model.

2.2 Grammar of Means Templates

A means template is formulated according to the following grammar:

```
MEANS_TEMPLATE  ::= COMPULSORY_PART  |  COMPULSORY_PART '[' OPTIONAL_PART
                    ']'
COMPULSORY_PART ::= '<' event ',' resource ',' DIRECTION ',' A_OR_CE
                    '>'
DIRECTION       ::= 'from' | 'to' | 'in'
A_OR_CE         ::= agent | value activity
OPTIONAL_PART   ::= E
E               ::= E 'AND' T | T | COMPULSORY_PART
T               ::= T 'XOR' F | F | COMPULSORY_PART
F               ::= other_event value_activity | other_event resource ','
                    DIRECTION ',' A_OR_CE | '(' E ')' | COMPULSORY_PART
```

The compulsory part of a template[3] is represented by a 4-tuple *<Event,Resource,DIRECTION,Agent>*. This part may be followed by an optional part providing complementary information about the consequences of the compulsory part. In the compulsory part, `DIRECTION` indicates the direction of an *event*, thus enabling us to distinguish between different situations (e.g. a resource moving "from" or "to" an agent). Notice also that the "`COMPULSORY_PART`" introduces the possibility to combine templates.

2.3 List of Means Templates

The following list of nine templates follows the list proposed in [1]. It covers goals related to the acquisition, production, maintenance, or provisioning of resources for a business actor.

[3] In this paper the notation "Ti" is the abbreviation for "Template number i"

1. <start offering, *resource*$_j$, to, *customer*$_i$ > [(start using existing *value activity*$_k$ XOR start producing *resource*$_j$ in *value activity*$_k$ XOR start procuring *resource*$_l$ from *provider*$_p$) AND receive *compensation* from *customer*$_i$]

Template 1 is used to express an exchange of an economic resource between agents. The main part represents the offering of the resource. There are three possible consequences in this template and they appear before the exchange showing the origin of the resource that is exchanged:
(1) The resource is converted thanks to an existing value activity in the agent,
(2) The resource is produced thanks to a new value activity in the agent,
(3) The resource is obtained by an exchange with another agent (intermediary).
After the exchange, the consequence is that a compensation is offered to the main actor for the resource provided.

2. <stop offering, *resource*$_j$, to, *customer*$_i$>
[(stop procuring *resource*$_l$ from *provider*$_p$ XOR stop producing *resource*$_j$ in *value activity*$_k$)]

Template 2 is used when an agent desires to stop offering a resource to another agent. The two possible (mutually exclusive) consequences are:
(1) The agent stops producing the resource,
(2) The agent stops procuring the resource from an intermediary.

3. <start procuring, *resource*$_j$, from, *provider*$_p$>
[((start using *resource*$_j$ in *value activity*$_k$) XOR (start offering *resource*$_j$ to *customer*$_c$)) AND start providing *compensation* to *provider*$_p$]

Template 3 is to express how the main actor deals with the procurement of a resource from an intermediary agent. The two possible (mutually exclusive) consequences of the acquisition are:
(1) The agent transforms (or use) the resource in one of its value activities,
(2) The agent offers the resource to another agent (or to the customer of the main actor, without changes).
After that, the main actor provides a compensation to that agent.

4. <stop procuring, *resource*$_j$, from, *provider*$_p$>
[(stop offering *resource*$_l$ to *customer*$_i$) XOR (start producing *resource*$_j$ in *value activity*$_k$)]

Template 4 focuses on stopping the acquisition of a resource from an agent. There are two possible consequences (mutually exclusive). The first of is to consider whether the agent stops procuring from all of its providers or not.
(1) To offer the resource to its customer(s), the agent must start the production,
(2) The agent do not want to continue the offering of the resource, so the offering of the resource is stopped.

5. <start producing, *resource*$_j$, in, *value activity*$_k$>
[start offering *resource*$_j$, to, *customer*$_i$]

Template 5 shows the consequences of starting the production of a resource in a value activity. The only effect is that the main actor must offer the resource to other agent(s).

6. <stop producing, *resource*$_j$, in, *value activity*$_k$>
[(start procuring *resource*$_j$ from *provider*$_p$) XOR (stop offering *resource*$_j$ to *customer*$_i$)]

Template 6 shows the consequences when an agent stops the production of a resource in one of its value activities. There are two mutually exclusive consequences:
(1) Keeping on going with the offering to other agent, the main actor starts procuring the resource from another agent.
(2) The agent do not want to continue the offering.

7a. <increase *fraction* of production of, *resource*$_j$, in, *value activity*$_k$>
7b. <decrease *fraction* of production of, *resource*$_j$, in, *value activity*$_k$ >

Template 7 is to increase (7a) or decrease (7b) the capacity of the production of a resource. According to [1], this template and template 8 has normally no structural effects on the business model.

8. <insource *fraction* of production of, *resource*$_j$, in, *value activity*$_k$>
[(start producing *resource*$_j$ in *value activity*$_k$ XOR increase production of *resource*$_j$ in *value activity*$_k$) AND (stop procuring *resource*$_j$ from *outsourcer*$_o$)]

Template 8 shows what is happening when the production of a resource from a value activity is insourced. There are two possible consequences (mutually exclusive):
(1) The production increase in an existing value activity,
(2) A new value activity is introduced to produce the resource.

9. <outsource *fraction* of production of, *resource*$_j$, in, *value activity*$_k$>
[(stop producing *resource*$_j$ in *value activity*$_k$ XOR decrease *fraction* of production of *resource*$_j$) AND start procuring *resource*$_j$ from *outsourcer*$_o$]

Template 9 captures the consequence of an outsource of production. An outsource is leading to:
(1) The stopping of the production of the resource (if the outsource represents 100% of the production),
(2) The decrease of the production of the resource.
In both cases, the main actor must start procuring the resource.

2.4 Transformation Rules

One transformation rule is associated with each template. A rule has two parts called the primary and the secondary action. The primary action is related to the compulsory part of the template. The secondary action is related to the optional part of the template. When applied, both parts of the rule affect the design of the business model.

The elements of the secondary action are either possible precursors of an event (i.e., what is needed to enable the event in a compulsory part of a template), or the possible consequences (i.e., what is done after a compulsary event). In other words, elements of the secondary action can both trigger or be triggered by events.

For space reasons we present here the rule that is associated with the 1^{st} template and omit the rules associated with templates 2–9. Table 1 gives the rule associated to the template 1:

1. <***start offering***, resource$_j$, ***to***, customer$_i$ >
[(start using existing *value activity*$_k$ XOR start producing *resource*$_j$ in *value activity*$_k$ XOR start procuring *resource*$_l$ from *provider*$_p$) AND receive *compensation* from *customer*$_i$]

Primary action:

(a) IF actor customer$_i$ is not present THEN add the actor customer$_i$.
(b) Add one value exchange for resource$_j$ (in an existing or new interface)
  from the principal actor to customer$_i$.

Secondary action:

(c) Add a new value exchange from customer$_i$ to the principal actor
  (as compensation for the resource$_j$ offered by the principal actor).
  Connect the new value exchanges to an existing or new value activity
  of resource$_j$ within the customer$_i$.
(d) IF start using THEN connect to the existing value activity$_k$
  to the new value exchange.
(e) ELSE IF start producing THEN call T5 and apply associated rule.

$\sigma = \{$ T5.resource$_j$ /T1.resource$_j$, T5.value activity$_k$/T1.value activity$_k$ $\}$

(f) ELSE IF start procuring THEN
    IF Prev(T5) THEN connect value activity$_k$ to the value exchange of resource$_j$.
    Call T3 and apply associated rule.

$\sigma =\{$T3.resource$_j$/T1.resource$_l$, T3.provider$_p$ /T1.provider$_p$,
  T3.value activity$_k$/T1.value activity$_k\}$

Table 1. Example of rule : rule associated to 1^{st} template.

It is important to notice that templates may be combined with other templates. We also say that a template may "call" another template. For example, the template matching with the offering of a resource is able to call the template responsible for the production of the resource to offer. In the rules, a function

"Prev(P_i)" has been introduced to avoid the possible issues of redundant changes. For example, if a rule says that an actor should be introduced, then the function "Prev(P_i)" is used to ensure that, in case templates are combined, the actor is not introduced twice.

Variables in templates are substituted before application analogously to how substitutions are carried out in Prolog [8]. For a better readability, the method distinguishes the initial substitution (γ_i) from those done when an additional template in a rule (σ_i) is called.

The calls (or combinations) between the templates can be represented within a tree for better visualization. While traversing the tree, the rules are modifying the as-is business model. Arriving at the right-most leaf of the tree (the final node), the as-is business model will be completely transformed and will be aligned with the goal model. Figure 1 shows an example of a substitution tree.

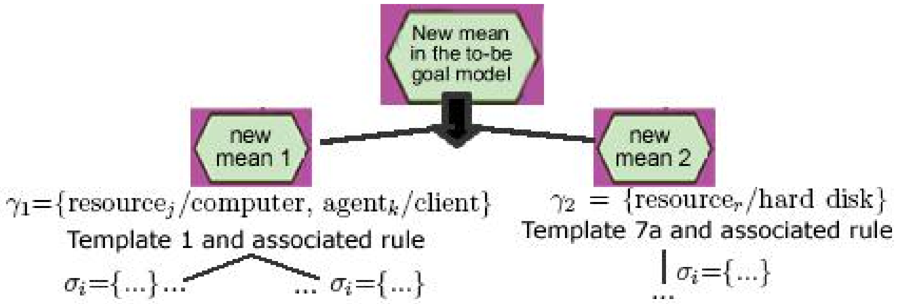

New mean in the to-be goal model
new mean 1
new mean 2
γ_1={resource$_j$/computer, agent$_k$/client}
Template 1 and associated rule
γ_2 = {resource$_r$/hard disk}
Template 7a and associated rule
σ_i={...}...
... σ_i={...}
σ_i={...}
...

Fig. 1. Through the goal model to the templates with a substitution tree.

2.5 Relations Between Templates

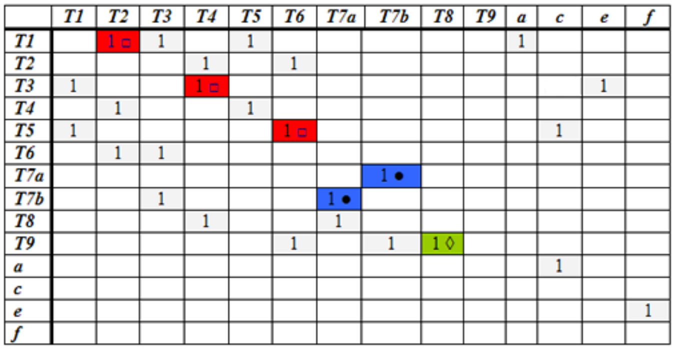
Consistency checking of combinations of templates is important. In an automated process, the matching templates found by the goal modeler in the "to-be goal model" have to be consistent. For instance, a business modeler must not declare a <`stop offering, resource, to, agent`> without a previous <`start offering, resource, to, agent`>.

In order to address consistency the notion of scheduling conditions is introduced. A scheduling condition is an expression of a particular combination of templates on which precondition must be checked to ensure its legality. A list of such conditions is highly context dependent. For instance, one combination may be allowed in one organization, while the same combination in another organization is forbidden. It is, however, interesting to sketch out and give an example of how one such listing can be done.

A scheduling condition has two parts. The first part is the combination part represented by two template symbols together with an infix composition operator "∘". This operator is used to express that a particular template can be combined with another. The second part is the conditional which contains a guard expression. Following is an example list of scheduling conditions. The first item, for example, express that template 2 can be combined with template 1 when the resource of T1 is equal to the resource of T2 and the agent of T1 is equal to the agent of T2.

$\mathbf{T}_i \circ \mathbf{T}_j$	**Precondition**
T2 ∘ T1	T1.resource=T2.resource and T1.agent=T2.agent
T4 ∘ T3	T3.resource=T4.resource and T3.agent=T4.agent
T6 ∘ T5	T5.resource=T6.resource and T5.value activity=T6.value activity
T7a	production(T7a.resource)≤100%
T7b	production(T7b.resource)≥0%
T8 ∘ T9	T9.resource=T8.resource and T5.value=T6.agent

The scheduling conditions may be organized and visualized in an "implication array" (Fig. 2). Template names are on the vertical and horizontal axes of the array. Implication dependencies are indicated by putting a symbol in the intersecting cell of a template column and row. Three symbols are used to indicate the implication direction[4]. A square indicate that the direction is from vertical to horizontal, a diamond from horizontal to vertical, and a bullet for a combination of both directions.



	T1	*T2*	*T3*	*T4*	*T5*	*T6*	*T7a*	*T7b*	*T8*	*T9*	*a*	*c*	*e*	*f*
T1		1 □	1		1						1			
T2				1		1								
T3	1			1 □									1	
T4		1			1									
T5	1					1 □						1		
T6		1	1											
T7a								1 ●						
T7b			1				1 ●							
T8				1			1							
T9						1		1	1 ◊					
a												1		
c														
e														1
f														

Fig. 2. Implication array for drawing global network

When drawing the array, two kinds of call between templates can be written: the "explicit" or the "implicit" calls. The explicit calls are highlighted in grey scale in the array. They regroup calls to some templates made from a rule associate

[4] The implication symbols are chosen to allow for gray-scale printing: □, ◊, •.

to another template. For instance, the rule associated to T4 can call T2 and T5. For these explicit combinations, there is no need of scheduling rules because the calls are made from inside the rule and ensure in this way, the satisfaction of the precondition. For implicit calls, scheduling rules with preconditions need to be considered. In the implication array three implicit calls has been added and highlighted.

- A link from the start to the stop (because it is only possible to stop something that has been started before) (□).
- A link between increase from decrease (in both directions) (●).
- A link from outsource to insource (because it is only possible to insource something that has been outsourced before) (◊).

Notice that a,c,e, and f are not templates but "other_events" as mentioned in the BNF grammar and that the implications between those events are "informal". We call those implications informal as they are merely for expressing implications in a language more natural to use. An analysis of the links between the templates is interesting because it makes it possible to avoid redundant changes on the "to-be business model" within the templates, thanks to the Prev(Ti) function.

3 Case Study

In this section we illustrate the method by applying it in a small case study (adapted from [1]). Due to space constraints it is not possible to show all models. We will, however, detail four templates matched with the means of a to-be goal model, one application of a rule associated with one of those templates, and the final output.

3.1 The Case

The case involves a Massive Multiplayer Online Gaming (MMOG) provider as the main actor. In this kind of game, thousands of players can participate via Internet and compete with each other. Two other actors interact with the MMOG provider: an Internet Service Provider (ISP) playing the role of a business associate, and the players as its customers. The MMOG provider has mainly two responsibilities: producing the game content by itself and distributing the game client application on CDs. Thanks to the ISP, the MMOG provider can distribute the information needed to play via the Internet. The revenue model for the MMOG company is based on fees collected to get access to the game server. This payment gives the right to access to the game. Obviously, the players need to be connected to the Internet in order to play.

For future development of its business the MMOG provider plans to change its goals and add new activities to support them:

1. Easier distribution of CDs by outsourcing the production of CD delivery to a shipper.

2. Reduction of its cost of content creation by outsourcing 50% of the game content creation to users.
3. Reduction of story boarding cost by procuring game stories from customers.
4. Increase the number of users by offering free trial games.

3.2 Method Application

To apply the method the goal modeller first has to draw the to-be goal model by introducing the changes into an as-is goal model. In the new goal model, new means are highlighted, matched, and formulated according to the means templates. After that, the rules associated to these templates are applied on the as-is business model to produce an aligned to-be business model as output.

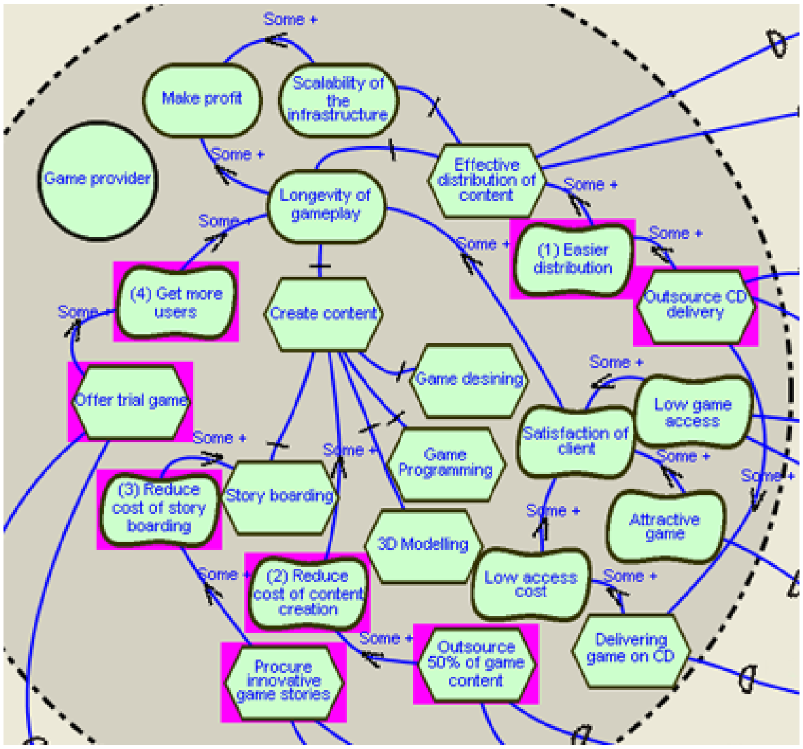


Fig. 3. Main part of the to-be goal model

To-Be Goal Model Figure 3 represents the to-be goal model of the MMOG company. This goal model was the as-is model that, when updated, became the to-be goal model. Eight new elements have been highlighted by rectangles.

Four of them are means (hexagonal) and four are so called soft goals (rounded rectangle).

New Means in the To-Be Goal Model From the MMOG provider's as-is goal model complemented with new goals according to the listing in the case description, we got the to-be goal model (Fig.3). Four new means are identified:

Means 1: Outsource 100% of CD delivery.
Means 2: Procure innovative game stories from customer.
Means 3: Outsource 50% of game content.
Means 4: Offer free trial games to customer.

Thanks to the usage of γ-substitutions, those means can match with some templates. The substitutions will have as effect instantiation of the terms: value activity$_k$, resource$_j$, ... The means 1, 2, 3, and 4 are respectively matched with the templates 9, 3, 9, and 1. The matching can be done thanks to substitutions chosen by the business modeler:

γ1={fraction/100%, resource$_j$/CD delivery, value activity$_k$/transport CD, outsourcer$_o$/shipper}

γ2={resource$_j$/innovative game stories, provider$_p$/customer, value activity$_k$/create content, compensation/payment}

γ3={fraction/50%, resource$_j$/game content, value_activity$_k$/create content, outsourcer$_o$/Customer}

γ4={resource$_j$/free trial game, customer$_i$/customer, value activity$_k$/distribute game, compensation/attention}

Table 2 shows the 9th template and the substitution matching with the 1st means.

9. <outsource *fraction* of production of, *resource*$_j$, in, *value activity*$_k$>
[stop producing *resource*$_j$ in *value activity*$_k$ AND start procuring *resource*$_j$ from *outsourcer*$_o$]
γ={fraction/100%, resource$_j$/CD delivery, value activity$_k$/transport CD, outsourcer$_o$/shipper }

Table 2. Matching means 1 with template 9.

Application of Rules Four rules associated with the means templates highlighted in the to-be goal model are used to transform the as-is business model into an aligned business model. Due to space constraints, only the rule associated the 9th template is presented here. Also, to save space, only the applied part of the rule is included in the following listing. In reality more rules than this one should be applied (one rule for each template).

Alignment of Means 1 (application of template 9 and its associated rule)

9. <`outsource` ***`100%`*** `of production of`, ***`CD delivery`***, `in`, ***`Transport CD`***>
[stop producing *CD delivery* in *Transport CD* AND start procuring *CD delivery* from *Shipper*]

Primary action:

a. IF actor *Shipper* not present THEN create actor *Shipper*.

Secondary action:

b. IF ((stop production) AND (outsourcing = 100%))
THEN call T6 and apply part (a) and (b) of associated rule
σ = **{T6.value activity$_k$/Transport CD,T6.resource$_j$/CD delivery}**
d. Call T3 and apply associated rule
σ = **{T3.resource$_j$/CD delivery,T3.provider$_p$/Shipper,**
T3.value activity$_k$/Distribute content}

Call of T6 and application of the associated rule:

6. <`stop producing`, ***`CD delivery`***, `in`, ***`Transport CD`***>
[`(start procuring resource`$_j$ `from provider`$_p$`) XOR (stop offering resource`$_j$ `to customer`$_i$`)`]

Primary action:

a. Delete the duality with the value exchange from the *Transport CD* concerning the *CD delivery* within the principal actor.
b. IF *CD delivery* is the only value object produced in the *Transport CD* THEN delete the *Transport CD*

Call of T3 and application of the associated rule:

3. <`start procuring`, ***`CD delivery`***, `from`, ***`Shipper`***>
[`(start using` ***`CD delivery`*** `in` ***`Distribute content`***`) AND (start providing` ***`Payment`*** `to` ***`Shipper`***`)`]

Primary action:

a. IF actor *Shipper* is not present THEN add the actor *Shipper*.
b. Add a new value exchange for the *CD delivery* from *Shipper* to the principal actor.

Secondary action:

c. Add a new value exchange from the principal actor to the *Shipper* (as *Payment* for the ***CD delivery*** offered by the *Shipper*).
Connect the new value exchanges to an existing or new value activity of *CD Delivery* within the *Shipper*
d. IF start using THEN connect the new value exchanges of *CD delivery* to the existing *Distribute content* activity.

The chain of templates called for the transformation of the as-is business model can be visualized as a tree. The first level represents the templates related to the new means templates indicated in the to-be goal model. The tree is then traversed in a pre-order walk: first the parent is visited and then the left child before the right child. Figure 4 shows the tree with each template called.

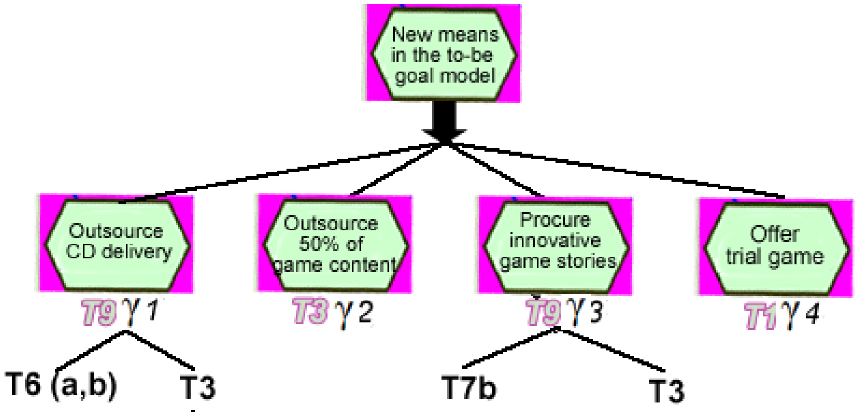

New means in the to-be goal model
Outsource CD delivery
Outsource 50% of game content
Procure innovative game stories
Offer trial game
T9 γ1
T3 γ2
T9 γ3
T1 γ4
T6 (a,b)
T3
T7b
T3

Fig. 4. Templates and calls from rules

Output: The To-Be Business Model Figure 5 represents the business model of the MMOG provider that is aligned with the to-be goal model. The figure is the as-is business model[5] where some transformations have been done by applying transformation rules. Newly added constructs in the model have been highlighted in the figure. The detailed example of a rule application resulted in the introduction of a new actor (Shipper) and constructs related to it.

[5] Notice that the as-is business model is omitted due to space restriction.

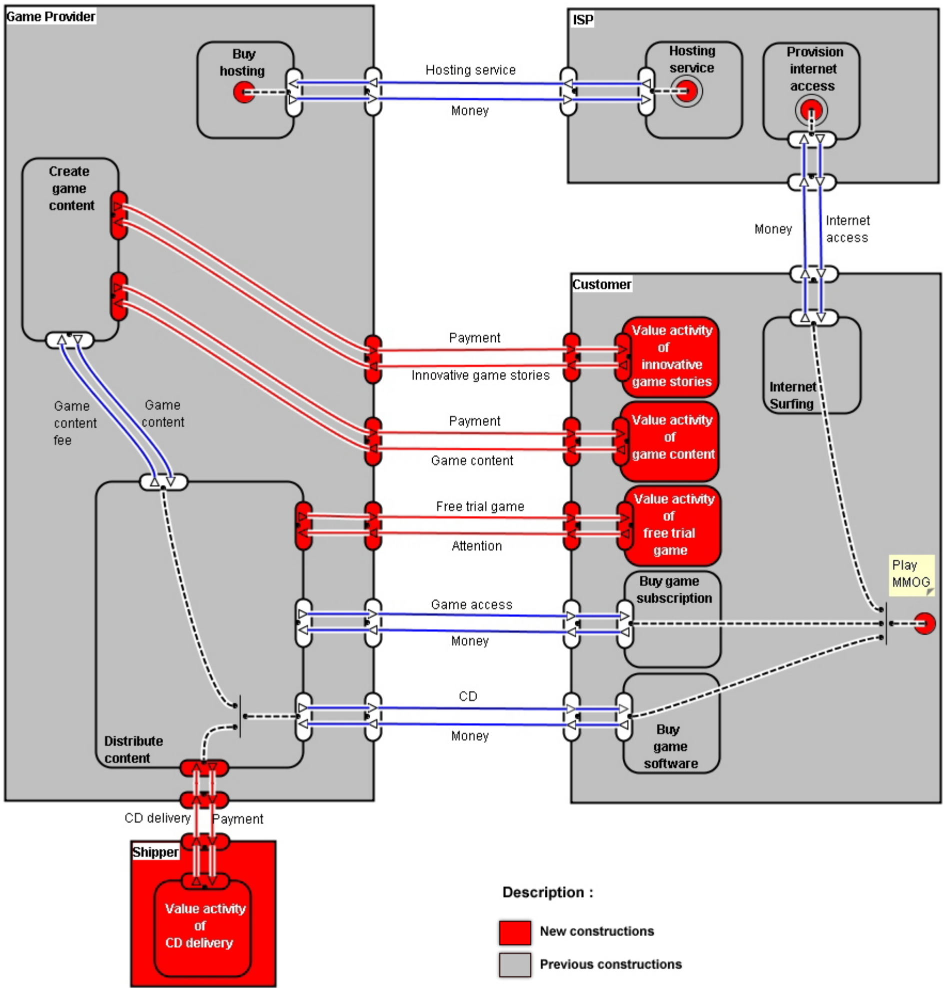


Fig. 5. Improved method : to-be business model

4 Conclusion and Future Research

In this paper we have addressed one part of the problem of business and IT alignment. We have done so by proposing a method that aims at aligning goal models and business models. The method takes as input a goal model and a business model, and outputs a business model that is aligned with the explicit goals of a business actor. The method builds on previous work presented in [1] and the benefits of that method still applies; clear and uniform goal formulations, well-founded business model design, and a high level of traceability. But this work extends the previous method in at least two ways. First, the syntax of both the the means templates and the rules was clarified resulting in better methodological support through reduced ambiguity. Second, combining templates to an arbitrary level is now possible. This enables a modeller to express whole chains of actions (or scenarios) that affect the design of business models.

Future research include investigations about the completeness of both the set of templates and the set of rules. For instance, as presented one template is always associated with one rule. This is very convenient as it constrains a modeller to arrive at only a small number of end results. It may, however, be that this is overly constraining and that more options should be open for the modeller. Another direction for future research is the proposed implication array. This array expresses the legal (or illegal) combinations of templates paving the way for consistency checking in the method. In order to do this checking the nature of the combinations must be understood. Some implications are always true, but some are true only in special cases. For instance, for a particular organization one combination of templates may be allowed while for another the combination is forbidden.

References

1. Andersson, B., Edirisuriya, A., Ilayperuma, T., Bergholtz, M., Johannesson, P., Zdravkovic, J.: On the alignment of goal models and business models. REA-25. A Celebration of the REA Enterprise Model, Geerts, G., University of Delaware. Department of Accounting and MIS. Available at http://www.aisvillage.com/rea25/program.html (2007)
2. Yu, E.S.: Modelling strategic relationships for process reengeneering. (1995)
3. Gordijn, J., Akkermans, J.M., van, J.C.V.: Business modeling is not process modeling, conceptual modeling for e-business and the web. LNCS 1921 (2000) 40–51
4. van der Raadt, B., Gordijn, J., Yu, E.: Exploring web services from a business value perspective. Requirements Engineering, 2005. Proceedings. 13^{th} IEEE International Conference on Volume Issue,29 Aug.-2 Sept. 2005 page(s) 53-62
5. Pigneur, Y.: e-business model ontology for improving business/it alignment, proceedings of the open interop workshop on enterprise modelling and ontologies for interoperability (emoi - interop'05). CEUR Workshop Proceedings (2005)
6. Gordijn, J., Petit, M., Wieringa, R.: Understanding business strategies of networked value constellations using goal and value modeling. In Martin Glinz and Robyn Lutz editors. Proceedings of the 14^{th} IEEE International Requirements Engineering Conference, IEEE CS, Los Alamitos, CA, USA (2006) 129–138
7. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T.: Strategic analysis using value modeling – the c^3value approach. 40^{th} Hawaii International International Conference on Systems Science (HICSS-40 2007). Waikoloa, Big Island, HI, USA. IEEE Computer Society. CD-ROM/Abstracts Proceedings (2007)
8. Nilsson, U., Maluszynski, J.: Logic, programming and prolog (2ed). Linköpings universitet, Department of Computer and Information Science, Sweden (2000)