

INFORMATION SCIENCE

BACHELOR THESIS (NO. 154)

Argument based modeling analyses on business process models

Author:
Rob THIJSEN

Supervisor:
Dr. Stijn HOPPENBROUWERS

June 27, 2011

Contents

| | | |
|----------|---|-----------|
| 1 | Problem statement, Background and Significance | 3 |
| 2 | Theoretical background | 5 |
| 2.1 | Business process models | 5 |
| 2.2 | Arguments | 6 |
| 2.3 | The anatomy of a split | 9 |
| 2.4 | Modeling practices | 11 |
| 2.4.1 | Modeling organizational processes in BPMN | 13 |
| 2.5 | A BPMN modeler | 14 |
| 3 | Research method | 15 |
| 3.1 | Subquestions | 15 |
| 3.2 | Operationalization | 15 |
| 3.2.1 | Comparing BPMN models | 15 |
| 3.2.2 | Grading a BPMN model | 16 |
| 3.3 | Methods of acquiring data | 16 |
| 3.4 | The arguments behind split types in BPMN | 17 |
| 3.5 | The experiment | 23 |
| 3.6 | Hypothesis | 23 |
| 4 | Results | 24 |
| 4.1 | Observations | 24 |
| 4.2 | Model scores | 25 |
| 5 | Conclusion | 26 |
| 5.1 | Argument classification | 26 |
| 5.2 | Answering the research question | 26 |
| 5.3 | Discussion: Future research | 26 |
| A | Modeling cases for the experiment | 28 |
| A.1 | Case 1: Grocery shopping | 30 |
| A.2 | Case 2: Doing the laundry | 30 |
| B | Resulting models | 31 |
| B.1 | Laundry case traditional | 32 |
| B.2 | Laundry case argument based | 33 |
| B.3 | Grocery case traditional | 34 |
| B.4 | Grocery case argument based | 35 |

Abstract

This thesis describes an attempt at formalizing a part of business process modeling. The research is focused on the AND & OR-splits in the business process modeling language. Several reasons for choosing such splits are given in hope of providing a standard recipe for choosing one split over the other for frequently occurring situations. The goal of this research is to formalize the AND & OR-splits in such a way that no doubt can exist over their meanings when a model is created, shared, and read. Business domain experts are helped by pre-abstracting information to the modeling world. A method for formalization of business arguments which are provided by experts in the field is used. After the process of argument formalization a short experiment is carried out using them. Finally a conclusion can be drawn about the quality of the formalized split arguments for improving model quality. In the future this might lead to automatic generation of business process models by only giving textual reasons. This is only possible if all and any reasons for a certain choice in the business process modeling language are formalized. A textual reason can be linked to a design choice because of this formalization, and then a model can be constructed by selecting the appropriate model translation of this design choice.

1 Problem statement, Background and Significance

With the following research is hoped to bring two worlds a little closer to each other. These two worlds are the worlds of information science and business administration. The world of information science tends to be more formal and structured than the world of business administration. Information scientists are constantly busy structuring information for (business) systems. Business administrators are busy running business and are less adept at formalizing information. Yet there has to be some form of communication between the information scientists and the business administrators. The business administration has to convey information and requirements to the formal world of system design. Without this communication it would be nearly impossible to design adequate information and process support systems for the business administration.

To bridge these two worlds one can make use of models. With the use of models, abstract information about business processes can be communicated. An information technology engineer can use this process information to design a supporting information system.

The models have to be in a shared “language” though. Shared between information sciences and business sciences. This means that both the sender and receiver have to understand what is communicated. When there is any room for ambiguity the communication can have undesired effects. A type of formalization has to take place to help the business side transfer more structured information. This formalization has to abstract the arguments that support different choices within a process to that they fit into a model. That’s where this research comes in.

The research question posed is:

| |
|---|
| <i>Can argument based modeling improve the generation of business process models?</i> |
|---|

This thesis is set in the environment of business process modeling. Why? Business process modeling has been around for a while and is a very basic way to describe workflows and information flows. Recker et al. [10] even claim that business process models are regarded as one of the most popular forms of conceptual modeling. This makes those models useful to convey information about business processes to (many) others. And most importantly: they are *business process models*. In the sector of information system design there exists a need for exactly this kind of information transfer. In order to write software to support a business process a system engineer has to know without ambiguity what the process looks like. The sharing of these models is of course not just limited to just information scientists and system engineers. Even for sharing process descriptions internally a business process model can be useful. See the figure below for a description of this communication process. Business process modeling is one of many choices, but its simplicity and versatility make it an interesting choice for the transportation of these data. By improving the ease with which business process models can be created, the quality of information sharing will improve as well. This makes this research quite significant in the field of information sciences (and perhaps others). Ways to improve model generation have practical applications.

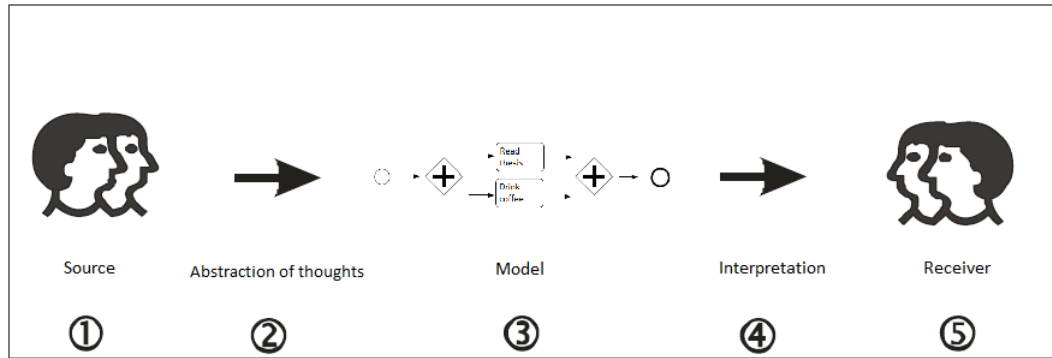


Figure 1: Model communication: via abstractions and perceptions [13]

Improvement on these models and their creation can be done in many ways. The sheer amount of literature about how to create a good model will support that claim. A part of the theory included is even a ‘way of working’ for creating a model. Currently the quality of a model depends on the facilitator, a person who is trained to process the information of a source in such a way that he can create the model for them. The facilitators are however not business domain experts. This thesis will focus on business arguments behind the models to ease the creation of good process models, instead of relying greatly on the facilitator. The arguments are the information pieces that matter. The model is just a way to convey abstractions based on this information. When these arguments are formalized correctly, it might be possible to construct an abstract model from them automatically. This would help the creation process of models tremendously. Automatic abstraction from business arguments helps gathering information on a level that the business domain experts understand best. Also this can mean there is no ambiguity in the information the model is representing.

Note that this process of formalization happens mostly on the business administration side of the field, as information scientists are already more used to working with formal models. One could say that the work is going from the business administration field towards the information science field. This is because business arguments taken from a domain expert at the business side are shaped towards a more formal description.

Having a predefined recipe for generating a model can have other uses than improving conceptual model quality generation alone. As stated in “A game prototype for basic process model elicitation” (Hoppenbrouwers & Schotten) [5] a well defined mapping of underlying arguments to a model structure has the following benefits:

1. Make lightweight formal modeling more accessible to non-expert modelers (i.e. enable modeling without the necessary presence of facilitators or expert analysts; this is sometimes called disintermediation)
2. Improve its quality
3. Improve its focus in view of its utility
4. Improve its efficiency
5. Improve the possibility to perform (formal) modeling in a truly collaborative setting

note: this list is cited from Hoppenbrouwers & Schotten [5]

In organizations complexity is ever growing. This leads to more complex models. These models are used for communication. Models that are easily created and improved on the points stated in the list above simplify and expand this communication, thus reducing errors. It’s therefore assumed that research in this area is significant.

2 Theoretical background

The research revolves around several key items:

- Business Process Models (BPMs)
- Arguments
- Procedures of process modeling
- Software tools

In this chapter the definitions of these and other theoretical keypoints will be given. Whenever possible the exact role of the concepts in the research is explained as well.

2.1 Business process models

The Business Process modeling Notation (BPMN from now on) consists of a set of visual elements that can be used to make a representation of an organizational process. The set of available elements is rather large, in this research only a subset is used. The BPMN and the subset are explained in this section.

A business process model is best explained as being a flowchart (White, 2004) [1]. This flowchart shows a hierarchical order of elements from a process. This process can be virtually anything: from the workflow of an assembly line to the flows of knowledge in a thinking process. The hierarchical order mostly models the temporal component, but could also express an iteration for example.

This flow model language has been standardized. The OMG group has assigned standard graphical elements to the process language. Now it is called the “Business process model notation”. This “Business process model notation” is designed as a high level business model language (White, 2004) [1]. The most basic operators in the BPMN are shown below.

A business process model can be made on different levels (White, 2006) [2].

- Process Maps: simple flowcharts of the activities
- Process Descriptions: flowcharts extended with additional information, but not enough to fully define actual performance
- Process Models: flowcharts extended with enough information so that the process can be analyzed, simulated, and/or executed

Each level has its own advantages and disadvantages when modeling certain types of processes. However, every modeling level can have its own level of abstraction. I.e. “simple flowcharts” can be constructed in great detail, visualizing every component of a process. It can also be constructed at a high level of abstraction, only showing core components. When modeling it is important to select the appropriate level of abstraction beforehand. More on that later. In this research only a subset of the available BPMN visual elements is used. Because of this the level of modeling is - contrary to the level of abstraction - fixed to simple flowcharts of activities in this research.

Activities Activities are atomic operations that take place within the process that is being modeled.

Non-atomic or compound activities fall outside the scope of this research. An example of an activity is “put stamp on paper”.

Events Events are “things that happen”. These affect the flow of the process. They can be best described as triggers. Events initiate a process (“start-event”) but also influence the flow of a process by conditional splits or joins. There are three types of events: start, intermediate and end events.

Gateways Gateways are the splits and joins of the business process modeling notation. Gateways can have simple or complex splitting and joining rules. Gateways will be described in more detail later. Conditional complex gateways are outside the scope of this thesis though.

Connectors Connectors are exactly what the name implies. Connectors visualize the flow of the process from one element to another. There are several types of connectors. There are connectors connecting elements to each other, and for example connectors that link comments to activities.

note: the information in this list is paraphrased from “Introduction to BPMN” (White, 2004) [1]

The corresponding visual elements of the BPMN are shown in the figure below. These visual elements are general, slight modifications denote activity, event and split subtypes.

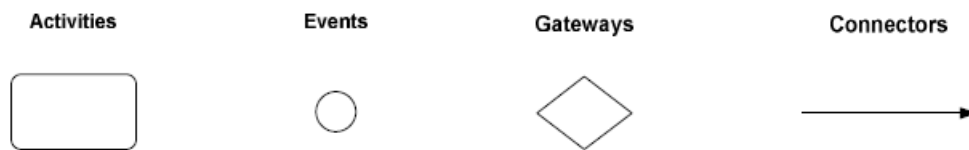


Figure 2: The corresponding visual elements of the BPMN

The other visual elements mainly allow for more advanced modeling of a process. For example, associations, swimlanes and message/dataflows are not mentioned in this list. Although there are more advanced elements in the BPMN element set, the four types of elements mentioned are the most important for modeling simple flows. Other components in the BPMN are mainly intended to depict flows of information rather than the flow of the process itself. In the scope of this research there is no focus on other elements as they do not have much relation to model splits and their reasons. With these four elements a process can be described on a basic level, it is however not feasible to build a model suitable for simulation. (White, 2004) [1].

The BPMN allows for the creation of junctions (the gateways). Junctions or gateways can be both splits and joins. These junctions can have various conditions. The junctions that receive the focus in this paper are the logical AND & the logical OR junctions.

The BPMN has a large role in this paper. BPMN will be used for applying argument based modeling on a model. BPMN was chosen because of its relatively simple structure and its widespread use. The list of arguments will be linked to certain model structures in the BPMN language.

For a more complete description of the BPMN the reader would best refer to the OMG tutorial to BPMN (White, 2006) [2].

2.2 Arguments

There are many definitions of the word “argument”. In this paper argument is a name for the very core information in the information models. Arguments are the reasons, logical constructs of the mind if you

will, behind decisions. Quoting the definition of argument from a dictionary:

ar-gu-ment

n.

1.

a. A discussion in which disagreement is expressed; a debate.

b. A quarrel; a dispute.

c. Archaic; A reason or matter for dispute or contention: "sheath'd their swords for lack of argument" (Shakespeare).

2.

a. A course of reasoning aimed at demonstrating truth or falsehood: presented a careful argument for extraterrestrial life.

b. A fact or statement put forth as proof or evidence; a reason: The current low mortgage rates are an argument for buying a house now.

c. A set of statements in which one follows logically as a conclusion from the others.

3.

a. A summary or short statement of the plot or subject of a literary work.

b. A topic; a subject: "You and love are still my argument" (Shakespeare).

4. Logic; The minor premise in a syllogism.

5. Mathematics

a. An independent variable of a function.

b. The angle of a complex number measured from the positive horizontal axis.

6. Computer Science; A value used to evaluate a procedure or subroutine.

7. Linguistics; In generative grammar, any of various positions occupied by a noun phrase in a sentence.

The meaning of the word argument in context of this thesis is found in definition 2. Arguments are reasonings and logic behind a certain fact or a certain choice.

A model is usually created by people with through and through knowledge about the process in question. These people are called domain experts. They have mental projections of how the process works. These mental projections have the form of arguments for a certain statement to be true. The *mental* reasonings of the domain experts are therefore the arguments for the *conceptual* model they are creating. A translation from the mental domain to the model domain is needed. This process is rather complex. Therefore only a short inaccurate description is used here: the process relies on abstractions made by the domain expert. Both from reality to mental projection, and from mental projection to conceptual model. Furthermore a modeling language has a level of abstraction from the reality it is describing. Not every aspect of reality can be built in the model without making it too complex to be of use (Campschroer) [12]. This means that there is a very large, complex, process of selecting what information to use and what information not to use. This process of translation between the mental and the model domain is difficult for people without the proper training. Usually a facilitator helps with this translation, or the facilitator may even make the whole model based on 'raw' information from the model creator. Formalizing arguments possibly eliminates the need for a part of that process by having pre-abstracted arguments of the mental projections from the business domain.

In other words: a person has a mental projection of how a process works. By using language the mental projection is converted to a conceptual model either with help of a facilitator or by speaking, writing, etc. Behind every fact in that person's model lies an argument, a reason for that fact to be true in the eyes of that person. These arguments are verbalizations of the mental projection. Not all information in these mental projections is needed for the model. Language forces a person to select a certain set of information. Argument based modeling may help in selecting what information is transferred.

When this translation from the argument domain to the model domain is automated, or at least formalized to some extent, the model will be a more accurate representation of the mental projection of that person. And, hopefully, be at the correct level of abstraction from reality. Although a domain expert may have valuable and precise knowledge of a process, he or she may however have difficulty making

an abstraction of the process. This is because *formal* abstraction is not a critical area of expertise for the business expert. Automatic or assisted abstraction using the formalized arguments helps resolve this problem. And therefore the model and its abstractions will be closer to the real situation that is being described.

This makes the arguments supporting conceptual models very important. Arguments capture knowledge. Why is there an event there? Why is this an AND split? The answers to these questions would be the arguments. When a model is used to give insight in a process, essentially an abstraction of this knowledge is presented. Because of the focus on AND & OR splits only arguments for these split types will be indexed and formalized.

2.3 The anatomy of a split

Splits and joins are forks in the process flow. These forks are called splits when one process flow splits into multiple flows. A fork is called a join when multiple flows combine into one. The splits and joins all have conditions for the way the process flow will continue. These conditions are often reflected in the name of the split or join. The splits are called “Gateways” in the BPMN. Gateways are used as both splits and joins in the BPMN modeling language. As this paper is focused on splits the term “split” throughout this paper, the term split is more often used than then gateway. When two process flows join, the term “Join” will be used. The distinction between these two operations is clearer then when only the term gateway is used.

To describe models, or particular elements in models one can use logic. One type of logic is better suited than another. Dijkman et al. [8] note that a logic to describe the BPMN should not be too rigid. However, as the formalization of splits takes place on the level of model elements, not on the level complete modeling languages and as it does not require anything else than a small set of rules combined with a motivation for them ordinary propositional logic is used. Because it is found that truth tables fit the split and join rule pattern best, logic derived from switching is used throughout the rest of this thesis.

When using a logic approach to describe splits and joins in a process flow model, one can compare these gateways with binary switching logic. The flow of the process can be compared to the flow of current in a circuit. There are inputs, an operation and outputs. There are active and inactive paths. For splits there is one input and there are multiple outputs. For join operation the reverse is true: there are more than one inputs and there is only one output. In the case of a logic AND split for example, an input can be assigned a binary variable A. The outputs are assigned variables X and Y. When a binary true is transmitted, it means the flow of the process flows following that path. When a binary false is transmitted, this means that the process does not flow along that path. When a flow is split over paths, which ones become active depend on the activities. Which ones *can* become active depends on the splitting rules. Now we can use truth tables to describe the behaviour of the process flow in the model over the splits (and joins). The operation that a split carries out is now a function from one variable - the input - to the others - the outputs. It is important to note that logic can describe the model flow, but is not flexible enough to described secondary requirements like time constraints or value constraints (Dijkmans et al. 2009). [8]

There are four types of splits in the BPM notation [2]. In this chapter the definition of splits is explained. The anatomy of these splits will be described. Differences and similarities will also be explained shortly.

The four types of splits in the Business Process modeling Notation are:

Exclusive gateways The exclusive gateway corresponds to the logical exclusive OR (XOR) join/split.



Figure 3: The BPMN visual element for an exclusive gateway

The truth table for an XOR operation is as follows (with ‘1’ meaning the flow follows the path of that output, and ‘0’ meaning that the flow does not follow the path of the output).

Table 1: Exclusive gateway truth table

| Input | Output 1 | Output 2 |
|-------|----------|----------|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |

Inclusive gateways The inclusive gateway corresponds to the logical OR join/split.



Figure 4: The BPMN visual element for an inclusive gateway

The truth table for an OR (inclusive) operation is as follows (with ‘1’ meaning the flow follows the path of that output, and ‘0’ meaning that the flow does not follow the path of the output).

Table 2: Inclusive gateway truth table

| Input | Output 1 | Output 2 |
|-------|----------|----------|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Parallel gateways The parallel gateway corresponds to the logical AND join/split.



Figure 5: The BPMN visual element for a parallel gateway

The truth table for an AND operation is as follows (with ‘1’ meaning the flow follows the path of that output, and ‘0’ meaning that the flow does not follow the path of the output).

Table 3: Parallel gateway truth table

| Input | Output 1 | Output 2 |
|-------|----------|----------|
| 1 | 1 | 1 |
| 0 | 0 | 0 |

Complex gateways Do not follow one set of logic rules. The complex split or join rules are defined by the modeler by use of events and other data in the model. This rule set is able to handle the many more complex types of data and flows in the model. The complex gateway is not found often in

the basic flow models that are subject to this research. For that reason the complex gateway will not be formalized.



Figure 6: The BPMN visual element for a complex gateway

2.4 Modeling practices

This chapter is dedicated to summarizing good modeling practices. Modeling practices are important because they will aid the process of formalizing arguments. The theory of creating good models helps in setting the right goals (in the form of requirements) for formalized arguments. The theory below is used for that quality goal setting in the process of formalization business arguments. Other than that, when argument based modeling is attempted one can make good use of a way of working. Even when using prefabricated arguments the modeling practices apply. While carrying out the experiment the modeling practices and the model requirements are in the back of the test subject's heads. Because of these reasons it is important to understand modeling practices before looking at formal arguments.

Jan Campschroer [12] has composed a paper listing most of the important modeling practices. Even when formalizing arguments to eliminate the need for a facilitator these practices are important to keep in the back of one's head. Because of their relevance to all approaches to creating a model, a summary of these design principles is given here. Argument based modeling has influence on most of these items.

In his paper Campschroer focuses on two main areas of attention. One strongpoint is that there are communication requirements for the information being modeled. The second strongpoint is that one has to have a way of working.

In the area of communication, Campschroer has pointed to the following requirements:

- *Understandability*

Understandability means that the persons that have to use model can understand it. This means that when the model is shared with people who do not have the same in-side knowledge as the creator, the model has to be created with the focus group in mind. The language and concepts in the model must be tuned to the readers of the model. This means that when arguments are formalized a facilitator or perhaps computer program has to modify the constructs of the model in such a way that they still fulfill this requirement. Argument based modeling will improve understandability by matching arguments with fixed pieces (constructs) in models.

- *Usability*

Usability in the words of Campschroer means that the model has the correct level of abstraction. All relevant parts of the real situation must be described in the model. The model must fit its purpose. The model must also fit its users. Usability is one of the grading point of models. Argument based modeling aims to increase the usability of models.

- *Falsifiability*

Falsifiability means that one can think of a situation that is not described in the model. If all situations fit the model description, then the model is considered to be general and informative. Argument based modeling does not improve falsifiability. This remains a task of the creator of the model.

- *Completeness and relevance*

Completeness and relevance also describe the level of abstraction. It has to be just right: the model needs to describe every desired piece of information that needs to be transferred. It must also not be too complex. The model should not do more than its goal. Argument based modeling aims to improve completeness and relevance.

- *Integrity*

The model should be a true representation of the process being modeled. The model should have an objective view of reality. Argument based modeling aims to improve integrity by giving a standard model construction for certain arguments.

- *Maintainability*

Maintainability is important for models that are used for longer periods of time. These models often need to be revised or adapted for a slightly new situation. When the model can be changed easily this saves a lot of time. Argument based modeling aims to help maintainability as the predefined formalized arguments split the model into building blocks that can be connected and disconnected with relative ease.

note: this list is a direct translation from a part of Campschoers table of contents. The descriptions are strongly based upon his explanation. [12]

The advised way of working is:

1. *Select goal and focus group*

By selecting a goal and a focus group the modeler is forced to think about what he/she wants to do with the model. A model that is not focused on the correct group of users will probably not have the desired effect. The model will then be a waste of time and other resources. A model without a goal is equally useless.

2. *Analyze the focus group* Before creating a model one also has to know what the focus group expects of the model. What does the recipient think is important? What view does the recipient have? The model will have to be fine-tuned to these criteria.

3. *Determine organizational embedding* With organizational embedding Campschoer means that a model must be created in such a way that it will be accepted. The model has to reach the right persons. The right persons also need to be able - and willing! - to understand the model. Otherwise the model will again not transfer information as desired.

4. *Confine the area of attention* When creating a model it has to be confined in three dimensions. This prevents a model from becoming overly complex.

The following three dimensions need to be confined:

- (a) Reality, i.e. What part of the organization, process, etcetera
- (b) View, i.e. from what aspects should the situation be modeled
- (c) Level of abstraction, what level of detail should be used

These choices can not be made implicitly as they make up a large part of the context of a model.

5. *Choose appropriate modeling language* For different situations, views and abstractions there are fitting modeling languages. It is important to use the right one in order to transfer as much information as possible. When choosing the wrong one problems like ontological deficiencies can arise. For more details on this topic see (Recker, 2010) [10]. In the BPMN there are three levels of modeling details, choosing one of these three can also be considered as choosing the appropriate modeling language. Because the modeling language in this research is already chosen it is assumed that BPMN is fitting for the situations being described.

6. *Structure the model* When writing a paper one has to think of a way to work towards a conclusion, fill in theoretical details, and so on. In the case of creating a model the same approach is needed, one has to think about how the information will be presented.
7. *Document the model* All choices, both about the model itself and the information being modeled, should be documented. This makes sure that one can backtrack decisions. Argument based modeling might help somewhat. When arguments are presented a software tool or facilitator can automatically log the arguments, thereby keeping track of decisions about the information being modeled.
8. *Check the model* As explained in comparing BPMN models (3.2.2) a model has three levels of communication. All need to be correct for the chosen goal and focus group.
9. *Check the effect* After the model is made and used for information transfer one has to check up on it. Have the persons using the model acted correctly after they interpreted it?

note: this list is a direct translation from a part of Campschroers table of contents. The descriptions are strongly based upon his explanation. [12]

When applying these principles to argument based modeling, a few things come to mind: Of course it is best if the creator of the model adheres to this workflow. Argument based modeling is just a tool. It does however not control the maker of the model him/herself. When this workflow is not followed the quality of the model will suffer.

With the replacement of the facilitator by a piece of software in combination with the domain expert the responsibilities change as well. Normally the facilitator will adhere to this workflow and keep an eye on the model requirements. As the argument based modeling technique reduces the role of the facilitator (or perhaps completely renders a facilitator unnecessary) *someone* or *something* else will have to stick to the model rules and guidelines.

Someone in the previous sentence probably means that the source of the information has to keep in mind these things him- or herself. This requires some extra expertise on the side of the creator. This is exactly the opposite of what argument based modeling aims to do.

Saying that *something* would mean that the modeling software used should keep an eye out. It is speculated that implementing the workflow via a piece of software should not be too troublesome. Monitoring the fulfillment of the communication requirements would a lot harder. This requires a high level of understanding about what is being modeled from the model checking software.

2.4.1 Modeling organizational processes in BPMN

When BPMN is used in an organization it is used within the context of that organization. An organization has made several strategy choices, and its processes mirror these choices. When a process is modeled using any language - in this case BPMN - traces of this strategy will be found in this model. This strategy can roughly be aimed at product leadership, operational excellence or customer intimacy (Treacy & Wiersema, 1992)[11]. Each of these choices carries different priorities for organizational processes. Other variables are generated from the organizational strategy. Such as certain choices for non-functional design elements. These also have a large influence on choices within the model. For example: an organization can also choose more than one strategy area in which they want to excel, and that leads to splitting processes up more so that a front end can focus on customer intimacy while the back office focusses on operational excellence. All these factors have tremendous influence on the choices that lie behind the models of a process. Therefore it is important to know the organizational context before attempting to construct a model from formalized arguments. It is very important to know how the modeling process takes place in the normal situations, i.e. when a consultant makes a model. For a certain process there are several correct models that describe it. There are however differences in how the 'non-functionals' of these processes are handled. Non-functionals are almost always "ilities" things like 'availability, maintainability, reliability'. Parts of the possible choices are given by the organization, other parts however are thought of by the modeler him(her)self.

The arguments a domain expert may have are influenced by all sorts of factors. His or her strategy and wish to optimize towards certain requirements influence the arguments that the expert has for certain choices. The list of arguments that will be formalized will have to span all possibilities in strategies to be of use for a wide range of processes. Because of its influence the possible strategies and following requirements are also a good point to begin in composing a list of important split arguments. The strategy is a very important guideline for an organization. It gives a goal and meaning to an organization. All its processes are therefore aimed at achieving those goals (Treacy & Wiersema, 1992)[11]. Before attempting to create - or expand - the list of arguments for certain formal model examples one should have knowledge of the strategy of an organization. When applying the found arguments this is even more so, as it is probably necessary to fine-tune these arguments so that they fit in the organizational non-functional requirements.

2.5 A BPMN modeler

In this research paper only one software tool is used. This tool is called Yaoqiang BPMN editor (author unknown).

The Yaoqiang BPMN editor is useful for BPMN model building. It has no form of automatic modeling built in. Interestingly certain often used split forms are present, but they do not resemble the work in this paper by a long shot. Yaoqiang will be used as modeling environment when facilitating the experiment. Yaoqiang was chosen because of a few advantages. For example its compatibility with many systems, as it is java based. Also the relatively intuitive interface means that it can be used by test subjects with normal knowledge of computer use. Last but not least is the fact that it is freeware published under the GNU license. For more information about Yaoqiang BPMN editor, see its project page at sourceforge. <http://sourceforge.net/projects/bpmn/>.

3 Research method

3.1 Subquestions

To reach an overall conclusion on the main research question posed in the problem statement, first a satisfactory answer has to be found to a number of subquestions. The following subquestions are chosen:

1. What are the arguments for choosing an AND or an OR-split in the BPMN?
2. Can these arguments be classified?
3. Are models constructed from formalized arguments equally or possibly more structured than models constructed in the currently normal way?

It is completely possible that the formalization of the arguments behind AND split decisions is difficult or impossible. If this proves to be the case the research will still follow the rest of the intended path. It is then attempted to make a semi-structured argument system with the intervention of an expert during the modeling sessions. Models are then generated with help of a human expert in stead of in a completely automatic way. This may still prove to be an improvement on the current way of business process modeling.

3.2 Operationalization

Not all subquestions need operationalization for them to be answered. The first two will be answered simply by attempting the process of compiling a list of arguments. Subquestion 3 however does need to be operationalized to be answered.

The research elements, indicators and variables are defined as follows:

- *Elements*: The subject of this research paper will be the BPMN model
- *Indicators*: The indicators for these models are complexity, usefulness and completeness
- *Variables*: The variables for the mentioned indicators are respectively:
 - *Complexity*: The number of splits and joins necessary
 - *Complexity*: The amount of other elements necessary
 - *Usefulness*: The grade a group of users of these models assigns in terms of usefulness for the model topic
 - *Completeness*: The grade a group of users of these models assigns in terms of amount of required information present in the model

3.2.1 Comparing BPMN models

When comparing models one should look at three levels of meaning. According to (Dijkman et al., 2009) [8] a model can be called the equivalent of another model if the model conveys the same message on all three levels. The three levels of meaning are:

1. Syntactic similarity where we consider the syntax of the elements
2. Semantic similarity where we look at the semantics of the elements
3. Attribute similarity where we look at the elements themselves

note: this list is a redacted version from the one Dijkman et al. used for similarity between elements in models

If it is needed that models created using the traditional facilitator based approach be compared to models created using the argument based approach are compared, they will be compared on these three levels of meaning.

- When the *syntactic similarity* is low, the process models describe different things. If you will, the overall ‘input and output’ of two models should be at least mostly equal to say they are alike. When two models are more than 10% different in their syntactic similarity they will not be compared to each other.
- When the *semantic similarity* is low, two models say different things. Even when having the same ‘input and output’ two models can describe a different process. Because only models describing the same process case are compared, this is not an issue.
- When the *attribute similarity* is low, the process models use different internal BPMN elements. In the experiment it is actually expected to find models with a lower attribute similarity while they still have the same semantic and syntactic similarity. This is because the argument based model will use predefined structures differing from the ad hoc created model constructions of a facilitator.

3.2.2 Grading a BPMN model

When grading a model without comparing it to another a different approach is needed.

Complexity is an important factor in the use of BPMN models. Complexity is a rather difficult concept as too little complexity means that not all relevant information is conveyed by the model. Too much complexity, however, means that the model is overclassifying the problem. This means that too much information is conveyed. Occam’s theorem dictates that the simplest model satisfying all requirements of the process model is preferred above a more complex one.

Usefulness is ofcourse directly linked to the fact that the generated models need a practical application for business administration experts. Usefulness is topic dependent. It can roughly be linked to both syntax and semantics.

Completeness is an important factor. As mentioned earlier on, models contain information. To bridge the gap between business administration and system development correctly these models are used to communicate information. When this information is incomplete or otherwise ambiguous problems can arise.

3.3 Methods of acquiring data

Each partial question has its own method of data gathering. Shown below is a numbered list with the data acquisition methods per subquestion. The numbers of the method description correspond to the number assigned to the questions.

1. Using the available literature, and by studying practical cases, induction and interviews with experts in the field a list of arguments for various split types will be composed.
2. Using literature about formal models, modeling practices and classes of arguments, an attempt to structure these arguments is made. No testdata is acquired, however the answer to this partial question influences the rest of the research.
3. Sessions to construct 6 to 10 models is held. The models will be created using a few cases. See appendix A. The models will be compared and graded using indicators from the operationalization and according to the methods also described in the operationalization 3.2. The scores of traditionally generated models will be compared to argument based generated models. When the scores differ significantly (5% or more) it is concluded that one approach or the other is better. The sessions will be held behind a laptop, that records audio and captures screen input.

3.4 The arguments behind split types in BPMN

This section will list the reasons that were found for using a specific split type (limited to AND & OR splits). This list was composed by induction, and by consulting experts in the field. My thanks go to Jan Campschoer for his assistance. The list is not explicitly sorted, but it is attempted to put the most used reasons at the top of the list.

The arguments are all very much related to (constraints on) dimensions, like time and costs. These dimensions correspond to the non-functional requirements of the process that is modeled. When applicable these dimensions will also be named in the description. These dimensions are not to be confused with the dimensions of the model itself. The BPMN standard forces the dimension of the model to be a rather abstract measure of time. When one task is placed after another, this implies that the first task takes place before the second, because of the various arguments listed below.

The descriptions will state if the argument is actor related, task related, or process related. An actor is, as the name implies, an acting party. It can be man, machine or both. The acting party is responsible for doing the actual activity named in the BPMN model. This answers research question two. Three categories of arguments can be made. There are other meaningful categories that possibly can be of use, but in this case a rather general approach is taken. Furthermore an example situation is given, from which the rules are converted to a split type. This example situation will then be presented in model form.

A task is the work that has to be executed. A task can consist of one activity, a process branch, or a complete process.

The process is the complete flow of tasks and corresponding actors. It is a structure that defines the order of tasks. An example can be found in the figure below. It is in dutch, as it is the same example as is used in the experiment cases later on.

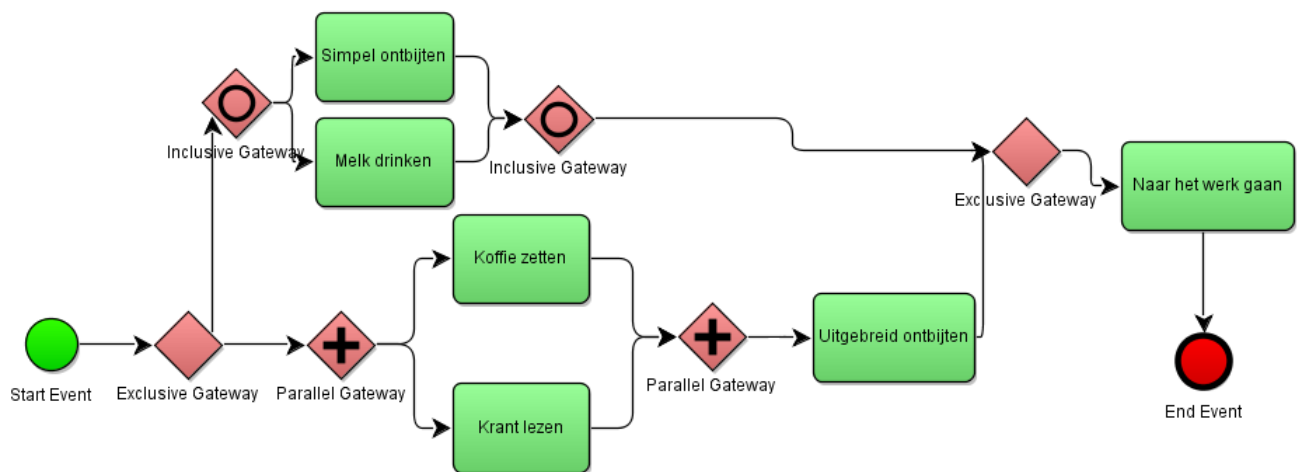


Figure 7: An example of a small BPMN modeled process, getting to work

Parallel processes Process related. In an organization, or any other workflow, a process is split into two tasks (or even subprocesses) that take place in the same timeframe. A parallel process is the result of two actors that have to do tasks. These tasks - in this case - are not completely identical, otherwise a split is not needed to visualize the way the process takes place. There might be a difference in activity or a difference in actor. The tasks have to take place in the same timeframe or at least in the same dimension the business process model is created. When tasks take place in the same dimension as the BPMN model and AND-split is needed. This AND-split then symbolizes an unconditional fork of the process. Of course the name already gives away what split is needed. Parallel process, parallel gateway. But when looking at the logic behind this, things become even clearer. In the desired situation the process should take two paths at the same time. But only when the process has arrived at the point where it requires parallelization. This means that only when the ‘input’ A (see figure) of the split receives the flow of the process, all of its output branches (in this case X and Y) should become active in the process as well. Otherwise all branches should be inactive. The fitting truth table for this situation is the ‘AND’ operation. See 2.3 for details.

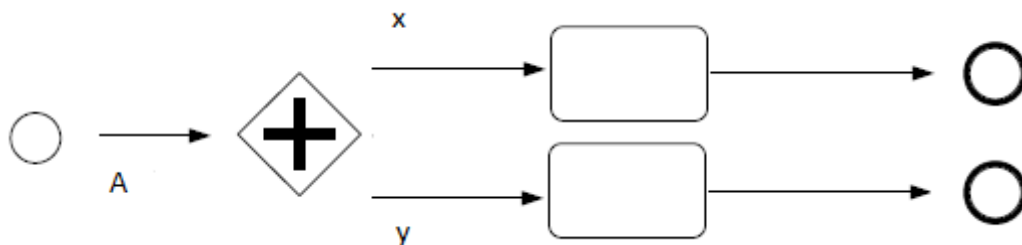


Figure 8: The BPMN model equivalent of parallel activities

Independent acting parties Actor related. Organizations almost always have a structure that organizes its workforce in departments. When a process involves two departments that have to do a part of the process - a task - at the same time, then an AND-split is needed. When only one departments has to do the task, but multiple are available, an OR-split is needed. This task can be the same for both of the two departments, i.e. “make a copy of a document and send it trough”, but it doesn’t have to be. This is different from the general “Parallel process”- reason! When a main process is split between two actors, the part of the process that each of the departments do can also be the same. When looking at the splitting rules for an OR-junction this fits the case: both branches X and Y of the process can be either active individually, or both at the same time. The only condition is that the input A of the split is part of an active process branch.

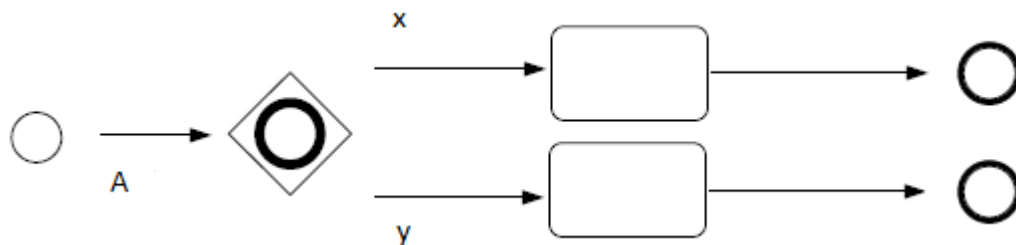


Figure 9: The BPMN model equivalent of independent activities

Dependency Task related. It might be the case that a process branch is dependent on the task of

the split before. In this case an exclusive-split is needed. The exclusive split should include the process flow as desired, and an alternative remedying the dependency. For example: when the answer to (and therefore the state of) a split criterion is either yes or no, the rest of the process will be dependent on the state of that answer. Given the answer only one task or process branch is followed. Because the answers are dependent on each other the AND-split is never chosen in this case. An exclusive gateway contains the right logic rules to create the situation needed: only when the gateway is in the active path of the process either X or Y can be part of the process flow at one time.

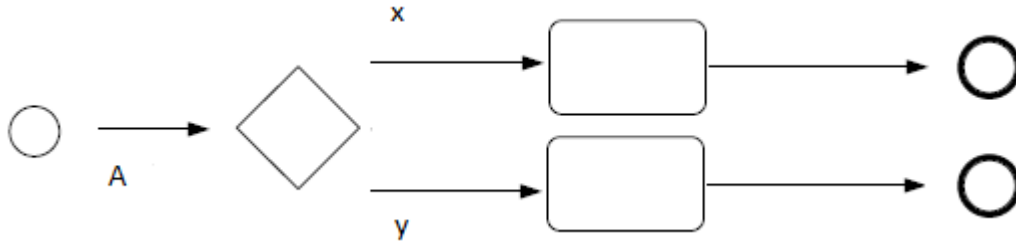


Figure 10: The BPMN model equivalent of dependent activities

Capacity enhancement Process related. Along the same line of thinking as splitting a task between departments is the reason capacity enhancement. Capacity enhancement is needed when a queue is forming at a certain task. This task is then called a bottleneck. When a processflow has a bottleneck, an obvious solution is to let more actors work at the bottlenecking task. This task therefore happens in two places at the same time. This situation requires an AND-split. This split will then behave as a parallel process. After the split the results are then often combined by an AND-join to again form one process flow.

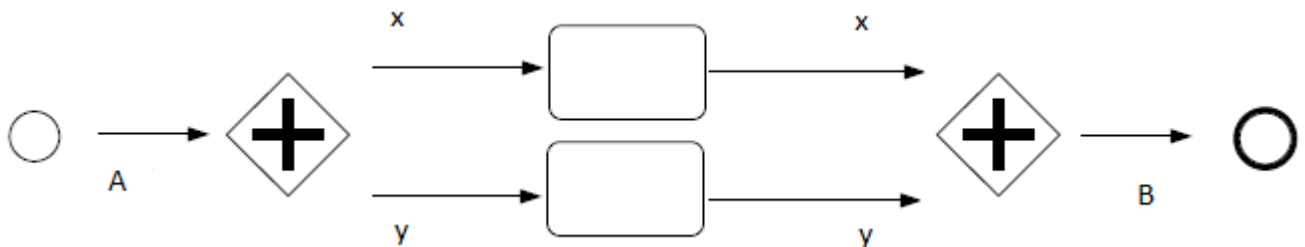


Figure 11: The BPMN model equivalent of adding capacity to bottleneck activities

Risk management Task related. When an organization has a very critical process that may not be disrupted, it is a good idea to make a split around blocking activities, so that there is a parallel process path circumventing it. This split is an OR split. When a critical part of the process is behind a certain processnode, it can be bypassed by an OR split, as the node mentioned is not critical. When the node is exactly the part of the process that is critical, it can be parallelized by creating a second identical part that has the same task. This combines the idea of parallel independent processes, capacity enhancement and different departments to some extent.

This means that there are two specific situations. In the case of a 'bypass' of a blocking non-critical activity the situation fits the truth table of an OR-split. Both branches X and Y of the process can be either active individually, or both at the same time. When both are active the bypass was not necessary, but does not disrupt the process either. The only condition is that the input A of the split is part of an active process branch. Note that because the blocking activity was non-critical the Y branch does not contain an activity. In order to complete the bypass both branches need to

be joined after the bypass is complete. This is an OR-join, as it follows the same rules as the split. Refer to paragraph 2.3 for the truth table of the OR operation.

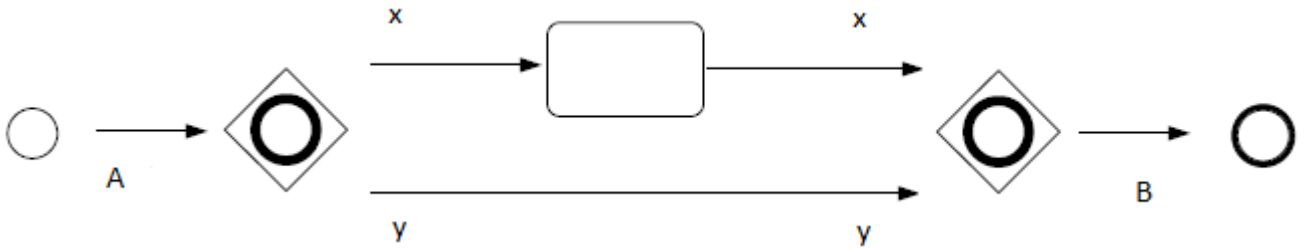


Figure 12: The BPMN model equivalent of a non critical bypass

This bypass construction does not work for critical process activities, as a step is skipped. In the case of a critical process activity an alternative should be ready. When branch X blocks, branch Y should have an alternative course of action. This is construction demonstrated in the figure below.

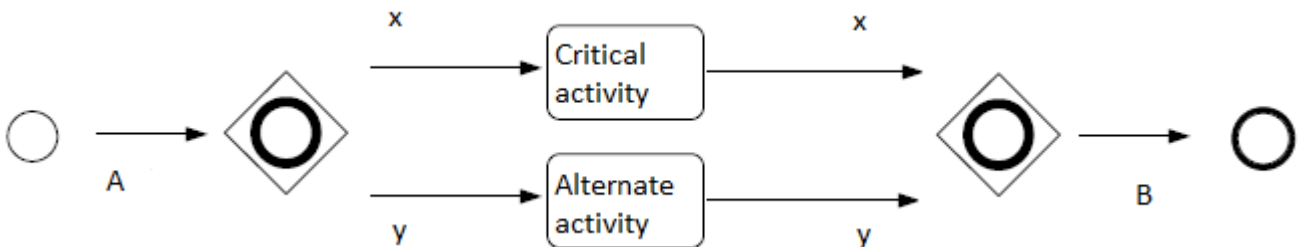


Figure 13: The BPMN model equivalent of a critical bypass

Competence Actor related. When some actors in an organization are better at executing a part of the modeled process, it is a good idea to create splits so that the most competent actor receives the specific task. When the extra capacity is not needed this construction should be avoided by not making the less competent actor part of the process. When multiple activities carried out by different actors are available an OR-split fits the situation.

In the following example, the activity in flow X and flow Y are equivalent. The first however is faster than the second one. It is however found useful to have the extra capacity of the less skilled actor for the activity on stand by in case it is needed.

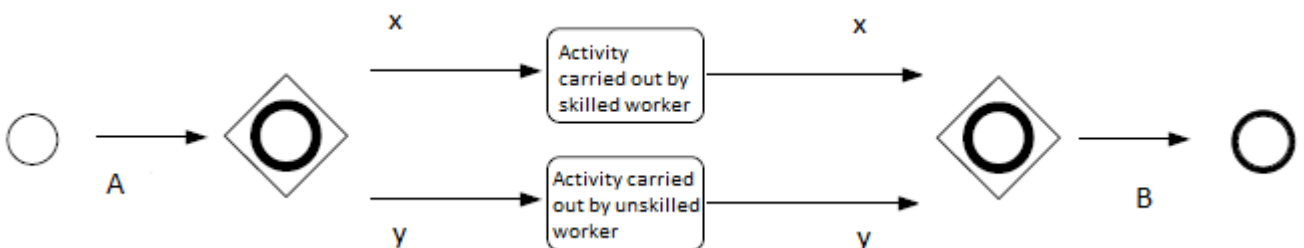


Figure 14: The BPMN model equivalent of a critical bypass

Cost effectiveness Process related. Cost effectiveness can lead to splits, but can also prevent them.

It can lead to splits in the same way as described in the “Competence” argument, it might be more cost effective to execute only one task of the process (XOR-split) in stead of both of them. Or the situation can be the other way around: possibly two different actors doing the same job (AND-split) can be more cost effective. In the other cases it can prevent splits. It might be not cost effective to use more than one actor or task, even when it is possible to use more. Then a split will not take place, while it is entirely possible to make one if there are different priorities.

When assuming that cost effectiveness corresponds to executing less tasks, an XOR-split can be used. Only one path is the active one, while the other is not active. When the longest path is inactive less tasks will be executed, saving money or other resources. An OR-split is not fitting for the task as it allows both the long and the shorter path to be active. That will result in higher costs, and the job being done twice.

Consider the following example. Path X may be suited for all situations, but be more expensive. Path Y may be good enough for certain situations, but can not cope with all. It is cheaper however.

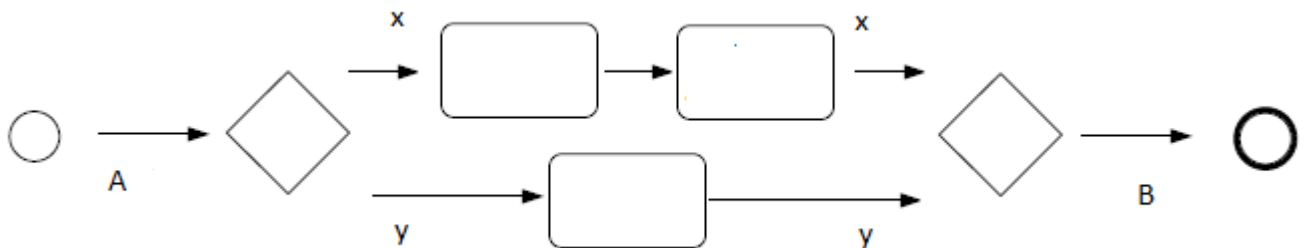


Figure 15: The BPMN model for a low cost bypass

Resource sharing Task related. An organization has resources which it uses to work towards a goal. To make good use of these resources they have to be divided. Ofcourse in BPMN this is done using splits. There are a few situations with accordingly different split situations. When one resource has to be shared amongst two or more actors, an XOR-split should be chosen. In that case only one actor can use the resource at the same time. When one or more resources are available but all actors can use them at the same time, an AND-split is the a better choice. In case of one resource being shared by multiple activities an XOR function fits the description of the situation. The XOR function lets only one activity be active at a single time.

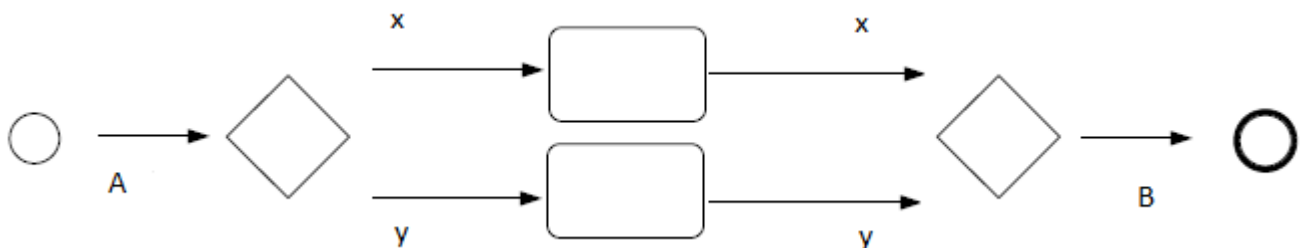


Figure 16: The BPMN model for a resource sharing operation

Employee satisfaction Actor related. It is possible that to optimize employee satisfaction a process with more or less splits are needed. Customer satisfaction is achieved by other means, for example by a high cost effectiveness, and is therefore not a part of the satisfaction argument. It is difficult to recommend one way of optimizing employee satisfaction by a split in a model. This differs per

organization and per process. Usually splitting a job over multiple actors lightens their load and realizes a higher satisfaction. This splitting situation is already covered by capacity enhancement.

To answer subquestion number two: it was found that there is an effective way for classifying the arguments for different split types in BPMN. These are mostly related to the non-functional requirements of the process that is being modeled. For most arguments, there is a pretty clear link to the type of split that should be used. However, not all arguments have a single split solution in the BPMN language. It seems that especially the fuzzy arguments like ‘satisfaction’ are not easily formalized. Quantifiable requirements are a lot easier to formalize into a logic truth table.

3.5 The experiment

The experiment aims to compare the traditional way of creating process models with the argument based technique described in this thesis. This will be done by modeling processes using both techniques, and by then grading the models. The experiment is held among students.

The experiment consists of the following steps:

1. The subject receives a short introduction to the BPM-notation
2. A case describing a process is given to the subject
3. The subject is allowed to familiarize him / herself with the case
4. The subject builds a process model using one of the two techniques
5. When several subjects have created models, they will be evaluated

Giving two cases to a subject with the purpose of letting them use both modeling techniques takes too much time. For validation purposes this would have been better. Both the ‘traditional’ and the ‘argument based’ model of one person could be compared. In stead different persons will make a traditional and an argument based version of a case. The cases describing a process are included in the appendices. The cases are day to day situations that can be described as processes. An attempt was made to make them complex enough so that most split types have to be used. At the same time the cases should not be too complex. When the cases are too complex the test subjects get confused, rendering their models useless for accurate comparison. Day to day situations were used so that the test subjects do not need to have much knowledge about a subject beforehand. This also helps proving that almost anybody might be helped by a formalized argument list when modeling a process.

During the experiment a facilitator will be present. In the case of traditional modeling, the facilitator will actively engage with the subject to translate the views of the subject to a model. In the case of argument based modeling the facilitator does not translate views to the model. When the argument named by the test subject is present in the formalized version, the model construct associated to it will be put in the model. When it is not, the facilitator will revert to the traditional approach. The models will be generated using the BPMN modeling tool described in chapter 2.5.

3.6 Hypothesis

The hypothesis is that a structured and formalized index of arguments for splits that is used in the modeling process leads to a better model as defined using the model quality indicators. This means that the expected complexity of the models generated will be less than the expected complexity of traditionally created business process models. The usefulness of the models might not differ widely. The completeness (and thereby correctness) of the models generated using the argument based modeling technique is expected to be better than the traditionally generated process models.

When the hypothesis turns out to be false, it is expected that the argument based models to be at least similarly useful and complete as the traditionally created models. An that creation of models is found easier than the traditionally generated models.

4 Results

4.1 Observations

The experiment to prove the validity of the chosen argument abstractions in chapter 3.4 was assigned a smaller role in the course of the creation of this thesis. This means that only two test subjects were part of the experiment. Both test subjects were males, in the range of 22-23 years old and both test subjects were students in the field of information science. The experiments yielded some interesting insights for me as the facilitator, but has limited practical and statistically relevant results.

Both students had difficulty understanding what was expected of them. If the time planning had allowed it, a more solid case description should have been written. Both student also tried to help me in the role of facilitator by making their own abstractions, cancelling out possible benefits of argument based modeling. Facilitating argument based modeling turned out to be less easy than expected. It is very important to have the prepared arguments at hand.

After these words of nuance I still want to share my experiences. Given the exploring nature of this paper, the lessons learned are of invaluable importance for any follow up experiments.

- During the session it became apparent that both test subjects had very different views on what should be modeled. This supports the idea that prepared abstractions can help in creating one unambiguous model of reality.
- It seems that argument based modeling saves some time, as the facilitator does not has to communicate back and forth a lot when the described situation fits a formalized argument.
- Both test subjects knew the logic rules behind split types, but still frequently hesitated at choosing which on to use. In a few situations they revised the model using a different split type
- The test subjects had difficulty using the arguments in the test cases. With the list presented, they agreed on the correct argument being modeled. They often deviated a little from its original pre-abstracted form. This indicates that the pre-abstractation helps them a long way, but falls a little short in some specific situations.
- Argument based models appear to have more structure than facilitator created models.
- Not all arguments are present to completely generate an argument based model. However, for the cases described, about 80% of the situations was covered.
- All of the models are different from the mental model I, the author, had when creating the cases. While creating the cases I thought of ways to put the argument based approach to use on them myself. I needed very little traditional modeling to create that model. The test subjects needed more traditional modeling to fill the gaps the argument based abstractions did not cover. This supports that the current list of arguments sometimes falls short, and needs expansion.

4.2 Model scores

The models that both persons created are attached in appendix B.

Grading the models lead to the following scores. The complexity numbers are gathered by counting the number of split elements and the number of activities. Arrows, start sequences and end sequences are not counted. Note that the grade assigned for usefulness is given by myself, the author of the case descriptions, and not as originally intended by a panel of students. As the author of the cases I am the domain expert so I feel my grades still carry some weight. Also, the theory about comparing models found in chapter 3.2 was used in establishing the completeness and usefulness qualities of a model in context of the rest of the models.

First off: the grocery shopping case.

| | Traditionally generated grocery model | Argument based grocery model |
|------------------------|---------------------------------------|------------------------------|
| Complexity: Splits | 10 | 18 |
| Complexity: Activities | 21 | 25 |
| Usefulness | 7 | 8 |
| Completeness: | 7 | 9 |

The argument based model has an increase in complexity on both the split and activity count. Despite this the models were found more complete, and more applicable for the case.

Secondly: the laundry case.

| | Traditionally generated laundry model | Argument based laundry model |
|------------------------|---------------------------------------|------------------------------|
| Complexity: Splits | 9 | 14 |
| Complexity: Activities | 17 | 14 |
| Usefulness | 7 | 8 |
| Completeness: | 6 | 8 |

The argument based model has an increase in complexity on the split count. There were less activities needed. Again the model was found to be more complete, and more applicable for the case.

Lastly the differences in scores between the two methods. The numbers for the complexity indicators for traditional and argument based models are the sum of all models. The grade points are also sums. The deviance is calculated by the summing the counts of traditional and argument based models, and seeing what percentage of that sum the difference between the counts is.

| | Traditional sum | Argument based sum | Deviance argument based from traditional |
|------------------------|-----------------|--------------------|--|
| Complexity: Splits | 19 | 32 | +25% |
| Complexity: Activities | 38 | 39 | +1,3% |
| Usefulness | 14 | 16 | 1/10 points |
| Completeness: | 13 | 17 | 2/10points |

It turns out there is a large increase in the amount of splits needed for the argument based models. The difference in this example is 25%! That is a lot more than the margin of error (5%). The number of activities differs somewhat, but not greatly. Less than 5%, so still within the margin of error. The argument based models are graded a little better when it comes to usefulness and completeness.

5 Conclusion

The answer to all three subquestions is found. The first two are answered by the included list of formalized modeling arguments. Although the list is not exhaustive, the often occurring arguments are included. Note that, in some cases at least, joins *and* splits have been formalized to give a more accurate representation of an argument in the model domain.

5.1 Argument classification

It turns out that it is very much possible to compile a list of formalized modeling arguments. The list generated in this research can be expanded. It is however not easy - or maybe even not possible - to make the arguments watertight. Some requirements of models are too fuzzy, immeasurable or complex to create a standard model element for a standard argument. The different level of abstraction between models contributes to this. The arguments themselves can indeed be classified in actor, task or process related.

The standard propositional logic was found to adequately cover the switching/splitting rules for formalized the gateways in the BPMN models.

5.2 Answering the research question

From the experiment results it becomes clear that the number of splits increases in stead of decreases when argument based modeling is used. The number of activities however did not change significantly. The other model quality scores favor the models generated using the argument based modeling technique. From this information is concluded that the extra splits in the light of the same amount of activities give a better but more complex structure to the model. Although more complex models are undesirable for several reasons [12], the increased structure is an improvement. The test subjects had to do little different - maybe even do less work - in the argument based modeling cases. The complexer models created with the same - or less - effort indicate that the pre-abstraction of arguments help the model creators deal with modeling situations. This suggests that argument based models are of better quality, being more structured, at the cost of added complexity because of a higher number of necessary gateways.

And that last conclusion leads me to believe that, yes, argument based modeling helps the generation of business process models. Only one note is needed: more research is needed to give a more statistically sound basis.

5.3 Discussion: Future research

The list of business arguments compiled in this thesis is far from complete. Expanding it using the same, or maybe even another method as is used here leads to more coverage of situations in the business world. The observed coverage of about 80% is too little to really exclude the need for a facilitator. There is also room for improvement on the current arguments by selecting a higher level of modeling. For example, by adding data flows and events thus choosing a larger subset of the BPM notation. Future research could thus focus on making the list more complete than it is, or focus on in depth model formalization.

There is also (much!) benefit in proving the use of the included arguments. Of course done by improving the experiment. The experiment carried out in this paper is of too small a scale to reach any solid conclusions. At least 20 test subjects and 40 models would give a much more significant result. The experiment can be improved by making the cases more watertight. In the ideal setup, the cases should be real world examples. The cases in this research are created with argument based modeling in mind and may therefore be more suitable for the argument based approach. The experiment can also be improved by giving the test subjects only arguments to choose from, and by then copying the corresponding formalized model parts in. This reduces the effect of the facilitator as a disturbing factor.

Lastly, when the list of arguments is more complete it is possible to look towards dialogue games

and argument based modeling working together. It might be possible to generate at least partly correct models from structured conversations.

A Modeling cases for the experiment

The cases and explanations are presented in Dutch, as the testpersons are from a Dutch university. All test subjects will be given one of the following explanations:

Bedankt voor je medewerking aan mijn experiment. Zometeen krijg je een kleine casus waarin een proces wordt beschreven. Dit proces zou je bekend voor moeten komen, maar neem de tijd om het geheel even door te lezen. Als je vragen hebt kan je die stellen.

Nu is het de bedoeling dat je met mijn hulp een flowmodel maakt van dit proces. Hiervoor gebruiken we de BPMN notatie. Dit houdt in dat er een beginpunt is. Deze wordt verbonden via de activiteiten in het proces naar een of meerdere eindpunten. Dit gebeurt doormiddel van pijlen. In het proces kunnen splitsingen plaats vinden. Er zijn drie typen splitsingen die je mag gebruiken: Uitsluitende (XOR) splitsingen, EN-splitsingen en OF-splitsingen. Ook moeten er processtromen bij elkaar komen. Daarvoor gebruik je ook splitsingen. Het is aan jou om de volgens jou juiste splitsing te gebruiken op plekken in het model waar jij denkt dat dit nodig is. Wanneer je wel een splitsing wil maken, maar niet weet welke, gebruik dan degene die je denkt dat het beste past. Ik zal zorg dragen voor het bouwen van het model, maar jij zult moeten aangeven welke elementen ik moet gebruiken.

Bij een EN-splitsing moeten alle in/uitgaande paden tegelijk waar zijn. Bij een OF-splisting mogen alle in/uitgaande paden gelijk zijn, maar ook een van beiden mag waar zijn. Bij een Exclusieve (XOR) splitsing mag slechts een van beide paden waar zijn, en de andere niet.

De symbolen voor dit model zijn als volgt:



Figure 17: Startpunt



Figure 18: Stoppunt

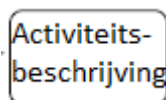


Figure 19: Een activiteit



Figure 20: XOR-splitsing



Figure 21: EN-splitsing



Figure 22: OF-splitsing

Hier is een voorbeeld van een BPMN proces. Jouw model hoeft niet helemaal te kloppen. Dit voorbeeld is vooral ter inspiratie.

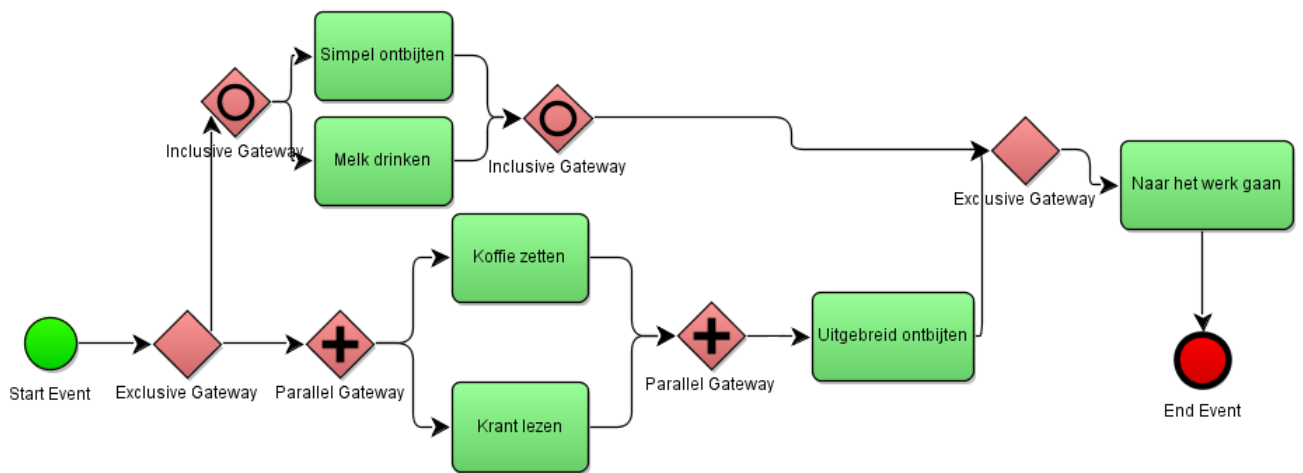


Figure 23: OF-splitsing

After this short introduction to BPMN the test subject were presented the following cases.

A.1 Case 1: Grocery shopping

Stel je voor dat je boodschappen gaat doen. Je bent thuis op je kamer en je weet nog niet wat je wilt hebben. Dit houdt voor jou in dat er een aantal activiteiten ondernomen moeten worden. Iedere keer als je boodschappen doet doe je de volgende activiteiten, dus alle mogelijke activiteiten moeten in je model zitten.

- Boodschappenlijst opstellen
- Je gaat met de fiets, auto, bus of je loopt
- Je kiest om de servicebalie op te zoeken voor vuilniszakken en dan te gaan winkelen, f juist de boodschappen te doen en dan de servicebalie te bezoeken
- De boodschappen op je lijstje zoeken. Soms neem je een dure variant, soms een goedkope
- Tijdens het boodschappen doen bel je met een vriend
- Er is een winkelmedewerker en die schakel je in om iets voor je te pakken
- Je besluit een product alleen te pakken als er niemand in de weg staat
- Er zijn twee kassa's waar je kan afrekenen. Je kiest een kassa die niet te druk is
- Je gaat naar huis
- Je leest je bonnetje door, maar alleen als dat bonnetje niet op dat moment gelezen wordt door een huisgenoot

A.2 Case 2: Doing the laundry

Stel je voor dat je de was gaat doen. Je deelt de wasruimte met een hele flat. De wasruimte heeft meerdere wasmachines en meerdere drogers Dit houdt voor jou in dat er een aantal activiteiten ondernomen moeten worden. Iedere keer als je de was doet onderneem je de volgende activiteiten, dus alle mogelijke activiteiten moeten in je model zitten.

- Je verzamelt je was uit de wasmand
- Je scheid je was in bont en licht. Soms zit er geen bonte was bij, soms geen lichte
- Als je veel was hebt gebruik je meerdere wasmachines
- Als je handoeken in de was hebt, voeg je extra wasverzachter toe, en anders niet
- Als je je was snel nodig hebt dan gebruik je de droger
- Soms heb je geen haast en dan gebruik je normale hangrekken. Er zijn er twee die je kan gebruiken, maar een werkt een stuk beter
- Je strijkt je was, maar alleen als je geen haast hebt
- Terwijl de wasmachines bezig zijn ruim je de was van een andere huisgenoot uit de droger
- Uiteindelijk vouw je de was op

B Resulting models

During the experiment four models were generated. There are two models per case description. Per case description a traditional and an argument based model was created. Audio (and partial video) recordings of the session are available on a sperate DVD.

B.1 Laundry case traditional

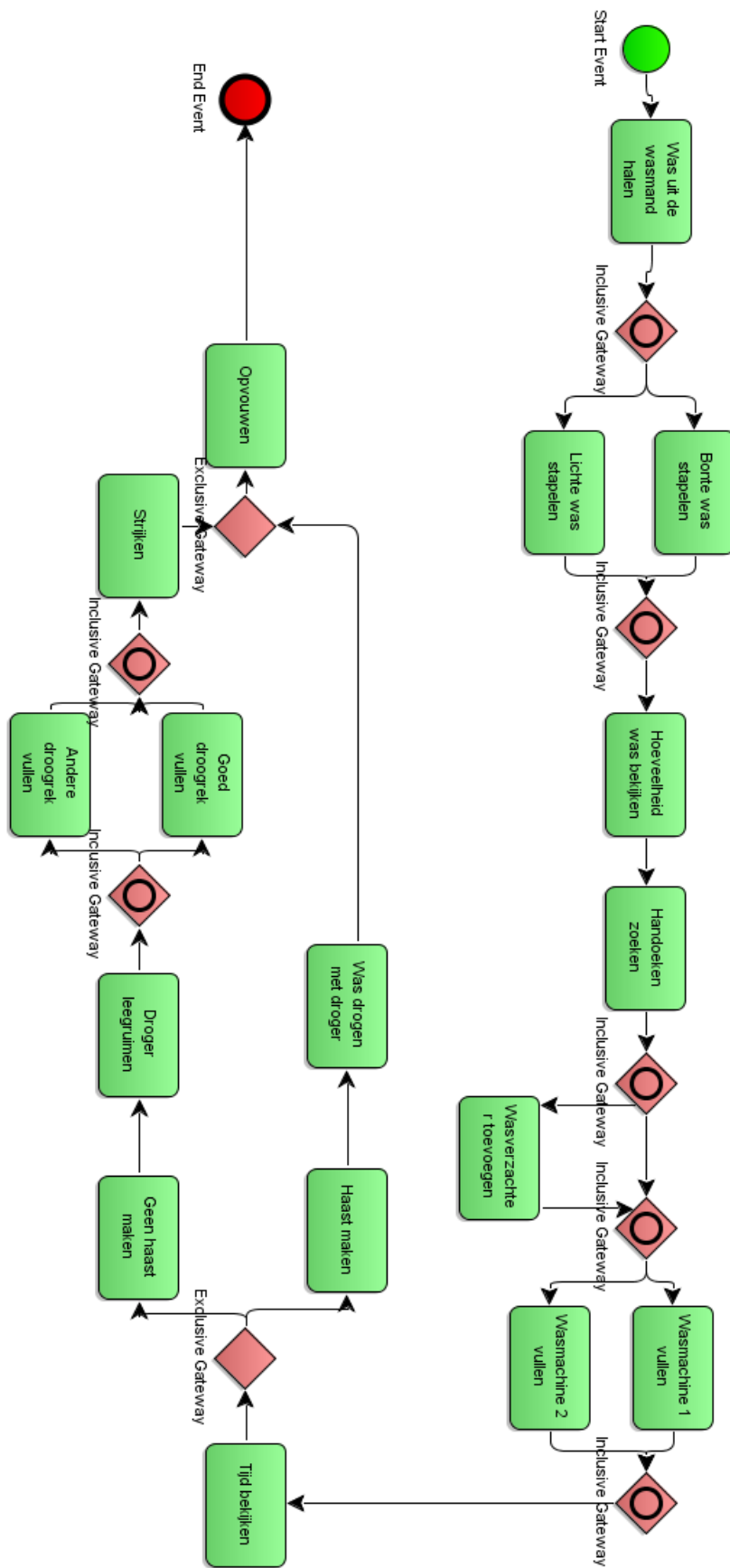


Figure 24: A traditionally generated model of the laundry case

B.2 Laundry case argument based

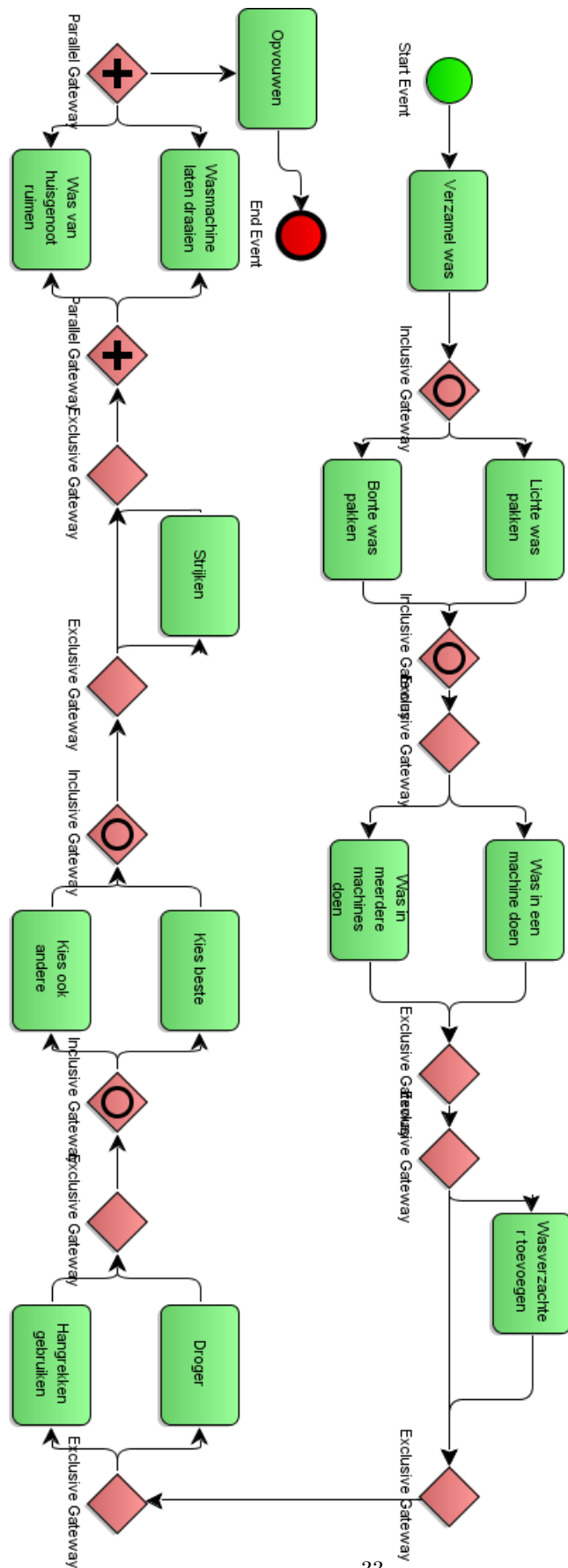


Figure 25: An argument based model of the laundry case

B.3 Grocery case traditional

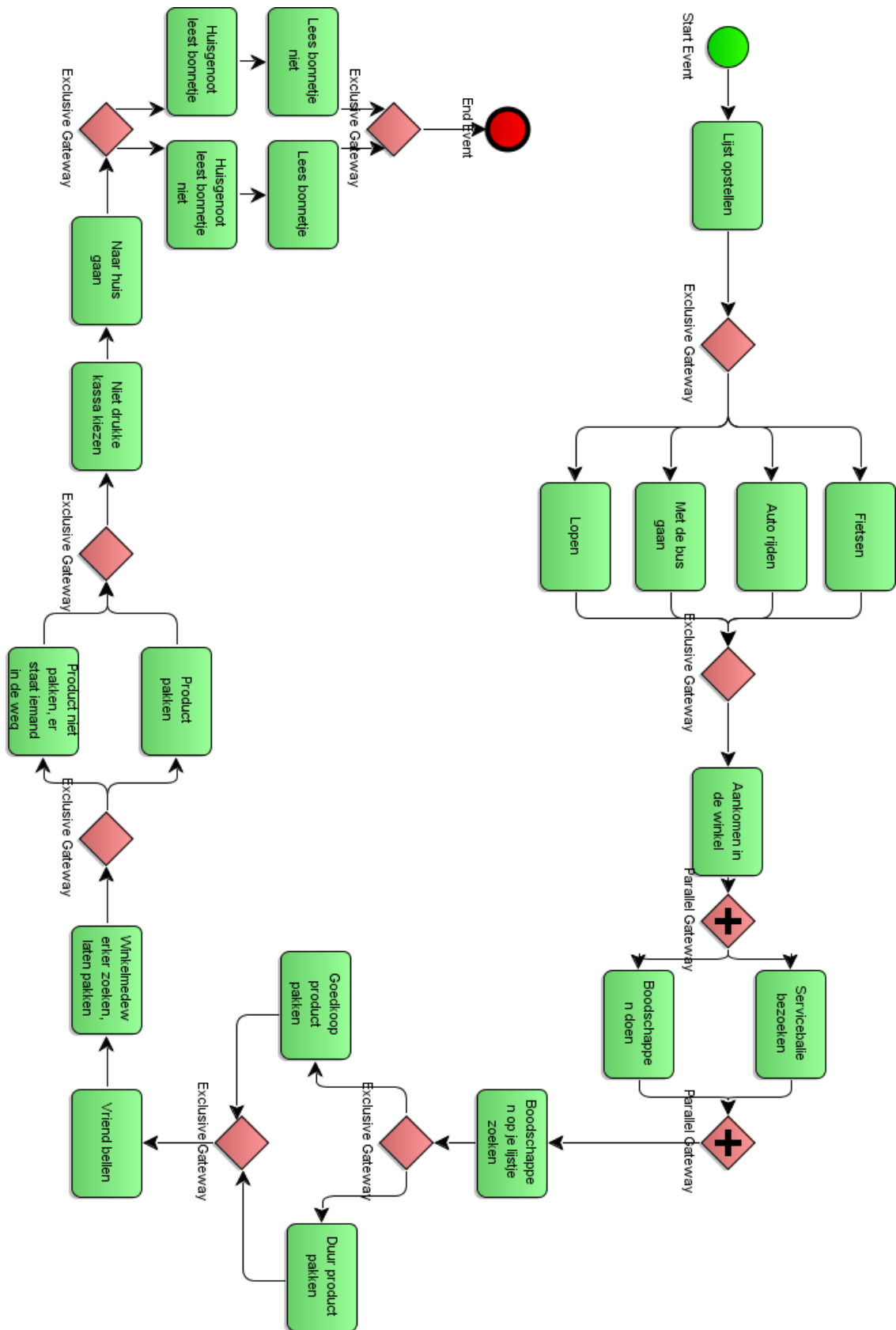


Figure 26: A traditionally generated model of the grocery shopping case

B.4 Grocery case argument based

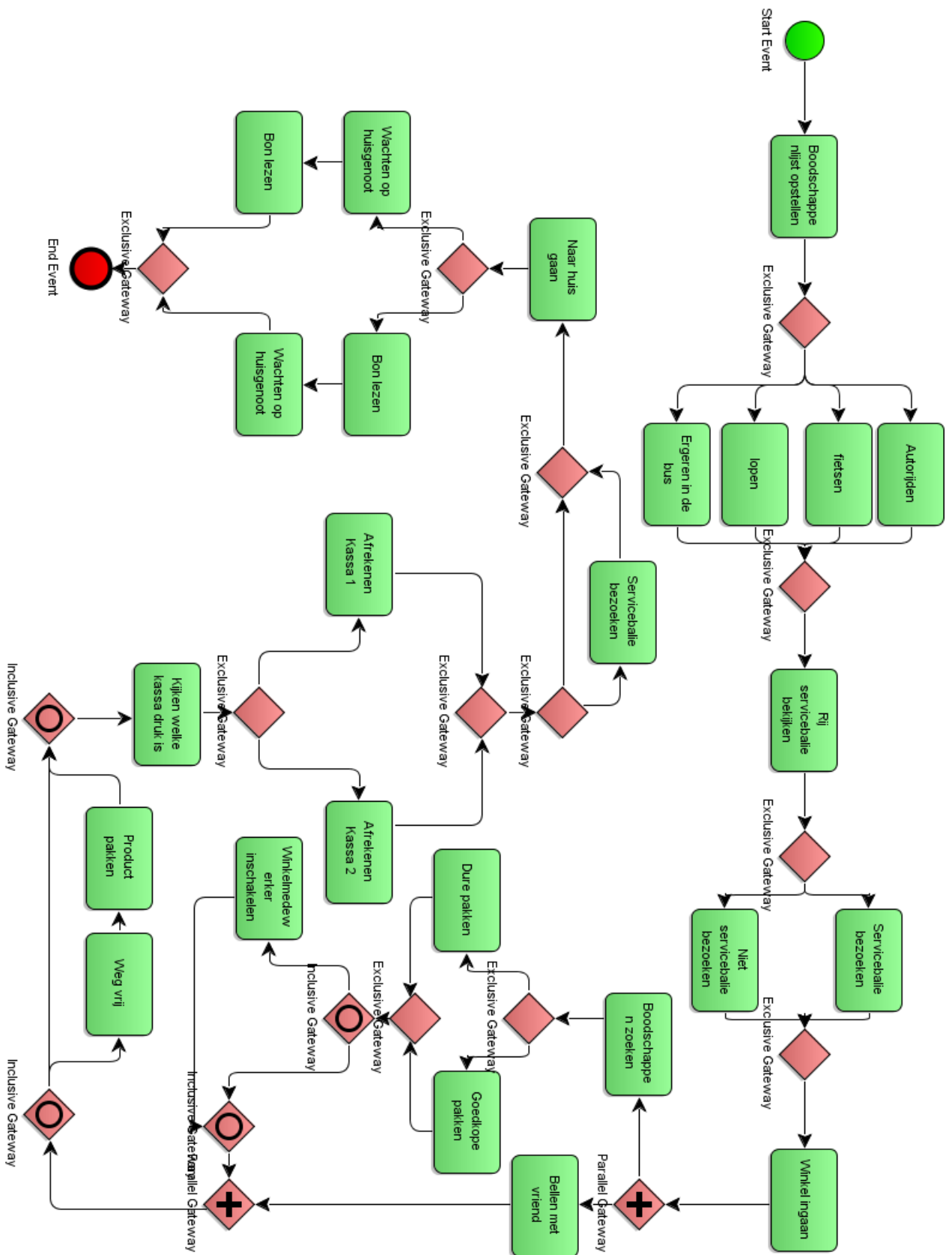


Figure 27: An argument based model of the grocery shopping case

References

- [1] Stephen A. White, 2004. *Introduction to BPMN*, IBM Corporation.
- [2] Stephen A. White, 2006. *Introduction to BPMN: a tutorial*, IBM Corporation.
- [3] <http://en.bpmn-community.org> at february 9th 2011
- [4] S.J.B.A. Hoppenbrouwers, H. Weigand, E.A.J.A. Rouwette, 2010. *Exploring Dialogue Games for Collaborative Modeling*, Radboud University Nijmegen.
- [5] S.J.B.A. Hoppenbrouwers, B.Schotten, 2010. *A Game Prototype for Basic Process Model Elicitation*, Radboud University Nijmegen - Institute for Computing and Information Sciences.
- [6] S.J.B.A. Hoppenbrouwers, B.Schotten, P. Lucas, 2011. *Towards Games for Knowledge Acquisition and Modeling*, Radboud University Nijmegen.
- [7] Egon Börger, Ove Sörensen, Bernard Thalheim, 2009. *On Defining the Behavior of OR-joins in Business Process Models*, Journal of Universal Computer Science, vol. 14, no. 1, Universities of Pisa, Italy and Kiel, Germany.
- [8] Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Käärrik, Jan Mendling, 2009. *Similarity of business process models: Metrics and evaluation*, Information Systems 36 (2011) 498516, Elsevier.
- [9] J. Mendling, B.F. van Dongen, W.M.P. van der Aalst, 2008. *Getting rid of OR-joins and multiple start events in business process models*. Enterprise Information Systems Vol. 2, No. 4, November 2008, 403419.
- [10] Jan Recker, Marta Indulska, Michael Rosemann, Peter Green, 2010. *The ontological deficiencies of process modeling in practice* European Journal of Information Systems (2010) 19, 501525.
- [11] Micheal Treacy, Fred Wiersema, 1992. *Customer intimacy and other Value Disciplines* Harvard Business Review
- [12] Jan Campschroer, *Communicatie met effect, Richtlijnen voor architecten van conceptuele bouwwerken* Ordina Consulting
- [13] John Fiske, 1982, *Introduction to Communication Studies*, London: Routledge