Chapter 4
Decision Activities

4.1 Introduction

Decision activities are independent actions of undetermined value that are carried out in the system. They are translated into the decision variables of the problem.

A decision activity is made up of the following components:

- The action that determines the activity: this corresponds to a verb.
- The elements that participate in the activity: elements of the system must participate in the action that corresponds to the decision activity.
- The quantitative meaning of the action: the quantification of the action defines the type of value of the variables. The meaning of an action can be of two types:
  - **Measure**: the result of the action is a value referring to any continuous measure (liters, kilos, time, etc.) or discrete measure (number of units) of an element. It corresponds to continuous or integer variables, respectively. If the measurement of units is bounded superiorly by one, the integer variable can be defined as binary. It is necessary that a measurable element (measurable individual or collective) participates in a measurement activity as object direct of the action.
  - **Logical**: the action corresponds to an activity of choice or selection for an element or group of elements where the response is evaluated with a logical value, True or False. The activity is determined with binary variables (1/0). The "1" corresponds to a positive or true evaluation and the "0" to a negative. All activities where only unitary elements participate will be logical.

The original version of this chapter was revised. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-57250-1_9

J. M. García Sánchez, *Modelling in Mathematical Programming*, International Series in Operations Research & Management Science 298
https://doi.org/10.1007/978-3-030-57250-1_4

The abovementioned factors are summarized in the definition of a decision variable. The definition of a variable must contain the elements that participate, the action, and the unit of measurement if the action is quantified as measure.

However, the identification of decision activities is not carried out in a singular way, but collectively or jointly. This occurs because the same action can be carried out independently by the participation of different groups of elements. The participation of each group defines a decision variable, but all the variables correspond to the same action in the system. Therefore, we will identify decision activities in the form of a set of events, in which an event is a valid association of elements participating in an action. The participation of the elements is unique with respect to the rest of the events. Each event corresponds to an individual decision variable regarding the joint decision activity.

In the definition of a variable, it is easy to identify errors in the design of the activity. You can detect inconsistencies, as a meaning without logic, an activity that has a known value and therefore is not a decision activity of the system, or simply a definition that corresponds to a function or calculation, for the wrong use of the participating elements. The variables generated from the decision activities must be independent. Their values are not obtained from any previous calculation.

Not all variables of a problem are independent. There are also the variables that store calculations. In them, the value is always obtained from the value of other variables, by calculating a linear function or a conditional function. The calculations of a system generate the calculation variables, which will be discussed in Chap. 5.

We shall now illustrate the definition of a decision activity and the analysis of inconsistencies without defining any rules, at least for now, in the definition of activities. We shall start with the first example.

Illustration 4.1: Production of Butter

A butter production factory wants to optimize its daily production of butter. Two types of butter are made (Sweet and Raw). A kilo of sweet butter gives the manufacturer a profit of $10 and a kilo of raw a profit of $15. For the production of butter, two machines are used: a pasteurization machine and a whipping machine. The daily use time of the pasteurization machine is 3.5 hours and 6 hours for the whipping machine. The time (in minutes) consumed by each machine to obtain a kilo of butter is shown in the following table:

Table 1. Butter processing times (in minutes)

	Sweet butter	Raw butter
Pasteurization	3	3
Whipping	3	6

Table of Elements (Table 4.1)

Decision Activities

The only action we can extract from the wording is “manufacture” butter, which is also used with the synonym “produce.”

Table 4.1 Elements of Illustration 4.1

Elements	Set	QN	Data				
			Name	Par	Type	Belonging	Value
Machines	$i = 1\ldots2$	I_U	Usage time	T_i	C (Min)	W	...
			Time consumed by 1 kg of butter j in machine i	TM_{ij}	C (Min)	S	...
Butters	$j = 1\ldots2$	I_M	Benefit	B_j	C ($/kg)	W	...
				TM_{ij}			

Action: PRODUCE.
Participating elements: The set of butters ($j = 1\ldots2$), that is, Sweet Butter and Raw Butter.
Quantification: The action of producing must obtain the quantity of butter produced since the quantity of butter is a property of undetermined value. This determines the measurable character of the butters. The unit of measurement used is the kilo.
Events: the activity generates two events:
Produce ⇒ Butter j=1 (Sweet).
Produce ⇒ Butter j=2 (Raw).
These events give rise to two **decision variables**:
x_1 = Kilos of sweet butter produced.
x_2 = Kilos of raw butter produced.
The information of the decision activities in simplified form would be:
Produce ⇒ Butter j = 1,2.
Decision variables: x_j = Kilo of butter type j produced.

If we analyze the definition of a variable, we can see the components (elements that participate in the event, action, and unit of measurement) that identify the activity.

Kilos	of sweet butter	Produced
Unit of measure	**Participating element**	**Action**

Incorrect Definitions

Let us consider some incorrect definitions we could have made of the activity. In Sect. 4.3.1, within the section dedicated to elements participating in a decision activity, and in Sect. 4.4 regarding quantification of the activity, we will present a series of rules that avoid these incorrect definitions.

1. *Suppose we identify a single event, in which the two types of butter participate:*
   Produce ⇒ Sweet Butter, Raw Butter
   A single variable is generated:
   x = Kilos of sweet butter and raw butter produced.

The definition of the components is correct, since it identifies the participating elements, the action, and the unit of measurement. However, the semantics could be understood in two ways:

A. We are assuming that the production of each type must be the same.
B. The activity represents the sum of the production of the two types of butter.

In proposition A, we make the mistake of assuming something that the system does not specify. On the other hand, the definition of a decision variable must not express a specification. The modelling of the specifications of a system has its own space that is carried out after the definition of activities. The decision activities must be defined independently, and then the specifications will impose the values they can take.

In proposition B, we are representing a function in the definition of the variable. We are adding up the production of the two types of butter. Therefore, we are defining a calculation on the correct decision variables, the amount of sweet butter and the amount of raw butter ($x = x_1 + x_2$).

2. *Suppose we incorporate the pasteurization machine into the participating elements.*

    Produce ⇒ Sweet Butter, Raw Butter, Pasteurization Machine
    For this action we define two events:
    Event 1: Produce ⇒ Sweet Butter, Pasteurization Machine.
    Event 2: Produce ⇒ Raw Butter, Pasteurization Machine.
    We analyze one of the events:
    Event 1: Produce ⇒ Sweet Butter, Pasteurization Machine.
    Variable: x_1 = Kilos of sweet butter produced in the pasteurization machine.

The semantics of the definition is correct and reflects all the components of the activity. The incorrectness is in considering the participation of the pasteurization machine in the decision of the activity. The kilos that are produced from sweet butter are equal to the kilos of sweet butter that are processed in the pasteurization machine. The machine does not contribute anything to the decision of how much to produce.

Kilos of sweet butter produced = *Kilos of sweet butter produced in the pasteurization machine*

Therefore, the participation of the pasteurization machine element in the decision can be suppressed.

When the element is implicit as a participant in all the events, it is not necessary to identify it, although there are occasions when, due to the clarity of the definition, it is maintained. In this case, it was not necessary.

3. *Suppose we identify the type of quantification as whole: the definition of the variables would be:*

    x_1 = N° of units of sweet butter produced.
    x_2 = N° of units of raw butter produced.

Obviously it is wrong because to define integer decision variables, there must be a collective element in the decision activity. If we take the unit as the kilo, we would be restricting the production to an integer number of kilos.

We are making the mistake of associating a discrete quantization with a continuous measurable element. The integer quantification is exclusively associated with the measurement of collective elements.

4. *Suppose we identify the type of quantification as binary: in this case the definition of the variables would be:*

$\alpha_1 = 1$ if we produce Sweet Butter; 0 if we do not produce Sweet Butter.
$\alpha_2 = 1$ if we produce Raw Butter; 0 if we do not produce Raw Butter.

This error is usually quite common in modelling. By defining the activity with this type of value, we do not have information about the quantity that we produce, and we will not be able to express the specifications and the objective of the problem. It is enough to analyze the data associated with the butters that always refer to the unit of measure of their quantity, both the production times in the machines and the profit.

The definition made of the activity, which obviously does not correspond to a decision activity but to a calculation, is a logical calculation. If we look at this in more depth, the values of α_1 and α_2 are obtained from the value of the correct decision variables of the activity ($x_1 =$ kilos of sweet butter produced; $x_2 =$ kilos of raw butter produced) via the following logical proposition:

If $x_1 > 0$ *then* $\alpha_1 = 1$; *if* $x_1 = 0$ *then* $\alpha_1 = 0$.

This would also be the case for α_2.

In the chapters dedicated to logical calculations and to specifications expressed as propositional logic, we will carry out an in-depth study of the use of conditional formulas and their modelling.

4.2 Actions of a System

The actions of a system correspond lexically with verbs (buy, sell, send, produce, install, assign, select, etc.) that may be accompanied by adverbs or prepositions. Incorporating the participating elements, to any action in a system it is possible to assign a type of value, either integer, continuous, or binary.

The actions with their participating elements that give rise to decision activities are actions of indeterminate value and are not always dependent on the value of other decision activities. Actions that do not fulfill these two properties cannot be considered as decision activities, either because they have a value already assigned or because they take values that are obtained from the values of other activities, that is, they always depend on other activities. These actions will result in system specifications.

As actions that can give rise to decision activities, those verbs that denote imposition (impose, limit, restrict, etc.) are excluded, since they are ways of defining specifications but do not define an activity in the system. Also, actions that can be missed out of the text because they act in an explanatory way, i.e., without any capacity to give rise to activities or specifications, are associated with a defined type of binary value. Let us take a look at an example:

Illustration 4.2
There is a system of buying a product from suppliers where you must encourage each supplier to buy at least 20 units.

Encourage is an action, but its value is determined; the wording clearly specifies that "we must encourage." This action could be eliminated from the wording without any problem, being as follows:

System of buying a product from suppliers where each supplier must buy at least 20 units

4.2.1 Actions with Calculated Value

These actions should not be considered as decisions because their value can always be calculated using other variables of the problem. The system allows to obtain the value of the action from the variables of other decision activities and even from other calculations.

If the actions of calculated value have been defined as decision activities, we must never forget the specification of the calculation that defines the action. That is why it is more advisable not to define them as an activity and to define them as a calculation, so as not to forget the restrictions that define it, as will be discussed in Chap. 5.

In terms of representing the calculation, the following will be involved:

Non-conditional Action
The activity value is determined directly by a linear mathematical function on other variables of the problem.

The actions with linear value will be part of what we will call auxiliary calculations.

Illustration 4.3
There is a system of buying and selling a product. To buy we have a set of three suppliers, and the products are subsequently sold on the market at a price of € p/unit. You must sell 50% of what you buy.

Buying is a decision activity. However, selling is not a decision activity because it can be defined as an auxiliary calculation:

$$\text{Selling} \equiv y = \frac{x_1 + x_2 + x_3}{2}$$

With x_1, x_2, x_3 decision variables of the activity of buying product from suppliers.

If the objective function manages costs and benefits, it will take the term py as a benefit.

Conditioned Action or Conditional Value Action

The definition of the action establishes the conditions to determine whether or not it occurs or the value it will take. The value of the action is determined by conditional propositions on other variables of the problem.

The conditioned actions will be part of the logical calculations and can be of two types:

– Conditioned action with a determined value: The action has an associated value. They are associated with binary variables.
– Linear value conditioned action: The value of the action is obtained from a function. They will be defined according to the value of that function.

Let us illustrate the two cases:

Illustration 4.4: Conditional Action of Determined Value

The system of Illustration 4.3 *has an activity to pay a fee of €100 if the units purchased exceed 200 units.*

The action would be Pay [a fee]. The fee would become an element of the system with an attribute of value equal to €100. The action of paying the fee is not independent; it is conditioned to the decision variables of buying. The value of the conditioned action is binary:

$$\alpha = \begin{cases} 1 & \text{if I pay the fee} \\ 0 & \text{otherwise} \end{cases}$$

The conditional proposition that defines its value is the following:

If the purchased units > 200 then I pay the fee, otherwise I do not pay the fee.
Mathematically: *If* $x_1 + x_2 + x_3 > 200$ *then* $\alpha = 1$, *otherwise* $\alpha = 0$.

The value of \$100 will be associated with the variable of the logical calculation α in the system cost function with the term 100α.

Illustration 4.5: Conditional Action of Linear Value

The system of Illustration 4.3 *has an activity of paying a fee of 1% of the total purchased if the units purchased exceed 200 units. The purchase price of the product is c \$/unit.*

The action is again Pay a fee, but in this case the fee does not have a certain value but the result of a linear function on the total purchased. Specifically, we can define the linear function as follows:

Price of the Fee $= 0, 01(x_1 + x_2 + x_3)c$

If we defined the conditioned action as binary, the cost function would have to be non-linear:

$$\alpha = \begin{cases} 1 & \text{if I pay the fee} \\ 0 & \text{otherwise} \end{cases}$$

If $x_1 + x_2 + x_3 > 200$ *then* $\alpha = 1$, *otherwise* $\alpha = 0$.

In the cost function of the system, we would associate the variable α with the function that calculates the fee $0, 01(x_1 + x_2 + x_3)c$, by means of the non-linear expression $0, 01(x_1 + x_2 + x_3)c\alpha$.

Since we always try to avoid non-linear expressions, the correct way would be to define the action of paying the fee with the value of the function that obtains the price of the fee, in this case a continuous value:

The logical calculation would be represented in a continuous variable z:

$z =$ Fee paid.

The conditional proposition that defines the calculation is:

If $x_1 + x_2 + x_3 > 200$ *then* $z = 0, 01(x_1 + x_2 + x_3)c$, *otherwise* $z = 0$.

In this way we will maintain the system with linear expressions.

Although we have entered into the logical calculations of a system, they will be examined in depth in Chap. 5.

4.2.2 Actions with Undetermined Value

Actions of undetermined value give rise to the decision activities of the system and therefore to the decision variables. The decision activities have values independent of the rest of the problem variables. Notwithstanding this, the specifications of the system can condition the values that the decision variables can take up to a point of being able to convert a decision variable into a variable of a calculated action. But a priori, without the specifications of the system, these values are free and are not decided by a calculation with respect to other variables. Let us take a look at a very simple example to illustrate this fact.

Illustration 4.6

In the purchase system of Illustration 4.3, *we define a specification that requires the total units purchased from the product to be 100.*

This imposes a constraint on the model of the form:

$x_1 + x_2 + x_3 = 100$

Table 4.2 Elements of Illustration 4.7

			Data				
Elements	Set	QN	Name	Parameter	Type	Belonging	Value
Stores	$i = 1...N$	I_U					
Raw material	–	I_M					

From this specification, the values of x_1, x_2, and x_3 are no longer completely independent, since $x_1 = 100 - x_2 - x_3$, $x_2 = 100 - x_1 - x_3$, or $x_3 = 100 - x_1 - x_2$. It could be understood that any one of these decision variables is an auxiliary calculation of the remaining two and therefore could have been defined as a calculation instead of as a decision activity. It is inevitable to find situations analogous to this, where the two interpretations fit. My opinion is that it is more structured to define them as decision activities and then represent the specification that relates them. Anyway, either of the two options leads to the same mathematical model.

On the other hand, decision activities can determine values of other decision activities, but for the latter to be defined as calculation rather than as a decision activity, the first decision activities must determine the value of the action for any of the values they take, without exceptions. If only the value of the action is determined for a subset of the values of the decision activities, the action retains its indeterminate nature and therefore prevails as a decision activity. Let's see a significant illustration of this fact:

Illustration 4.7

There is a system in which stores are rented and raw materials are stored within them. The system has a set of N stores that can be rented for storage.

The table of elements could be presented in the following way, taking into account the wording (Table 4.2):

Two actions are clearly identified, rent and store:

Action: Rent [stores].
Participating elements: Stores ($i = 1...N$ I_U).
Quantification: Binary.
Events: Stores $i = 1...N$.
Decision variables:

$$\alpha_i = \begin{cases} 1 & \text{if rent store } i \\ 0 & \text{otherwise} \end{cases}$$

Action: Store [raw material in stores].
Participating element: Raw material I_M; Stores ($i = 1...N$ I_U).
Quantification: Continuous.
Events: Raw material ⇒ Stores $i = 1...N$.
Decision variables: x_i: Amount of raw material stored in Store i.

Although not explicitly described, it cannot be stored in a store that you have not rented (the implicit specifications of a system will be discussed in Chap. 6). This establishes a partial determination of values between the two groups of variables:

If you have not rented Store i, you cannot store in store i:
$\Rightarrow$ *If* $\alpha_i = 0$ *then* $x_i = 0$.

One of the values of the Rent activity determines the value of the Store activity. However, for the value $\alpha_i = 1$, a value for x_i cannot be determined, so it does not determine x_i as a calculation.

On the other hand, activity x_i also partially determines the values of α_i, since if I have stored something in store i it is because I have rented the store:

$x_i > 0 \rightarrow \alpha_i = 1$

However, the missing value of x_i, $x_i = 0$, does not determine the value for α_i, because it could have rented the store and not stored any raw material.

Regarding the latter, it is quite reasonable to think, if there are no more elements in the system that must be taken into account to store in the stores or other activities related to the stores, that if we rent a store, with the supposed cost that this entails, it will be to store raw material. This has an important consequence: we can convert the activity of renting into an action of calculated value, and therefore it is no longer an activity decision of the system. In this case, we can contemplate the calculation of all values of α_i:

$x_i > 0 \rightarrow \alpha_i = 1$
$x_i = 0 \rightarrow \alpha_i = 0$

4.3 Participating Elements in a Decision Activity

Obtaining the right list of participating elements of an activity can be a complex task in some problems. In fact, it is the phase with the least ability to be regulated. Despite this, we will try to give some guidelines for a correct selection.

Indicating the elements that participate in an action requires full knowledge of the system and a table of elements that is defined correctly.

Obtaining the elements involved in an activity is based on looking for relationships between the elements of the system and the action. The search is based on asking to the action looking for as answer to the element.

The questions we can ask the action are dependent on the meaning of it. Questions about the value of the activity are logically excluded, like the question how much? Thus, they are generally questions of the type “What?”, “For what?”, “Who?”, “Where?”.

If the action is of logical value, obtaining the elements is usually simpler. You have to look for the elements that we associate with the election.

In general, if for example I have a Send activity, the logical thing is to ask What do I send? It will also seem logical to ask where do I send it from? And where do I send it? Asking the action is simply an informal tool to help identify the participating elements. The key is to have a rounded knowledge of the system and to fully understand the meaning of the activity.

4.3.1 Rules of Participation

Despite being an unregulated task, it is possible to define some work premises in the participating of the elements:

1. **An event of a measurable action can only measure a single element.**

In an event we cannot consider the quantification of more than one element since that would mean establishing a function, and a decision activity is an independent event, not the result of a calculation or function.

Illustration 4.8: Production of Butter
In our basic problem, the action we identified as a decision activity was PRODUCE. What do we produce? Sweet Butter and Raw Butter.

There are two measurable elements so we cannot measure both in the same event. It is necessary to generate two events:

PRODUCE Sweet Butter
PRODUCE Raw Butter

2. **In an action in which we identify the participation of an element that does not function as a direct object to the action, its participation combined or not with other elements should be an alternative or option among a set of alternatives, which will be collected in the events of the activity.**

There must be other elements that are alternatives to perform that action. If an element always participates in all the events of an action in a secondary way, we can exclude it because its participation is implicit. In spite of this, in some cases, to maintain a clear definition of the activity, we can keep it in participation.

Illustration 4.9: Production of Butter
As we have indicated in 4.1, with this standard we deduce that it is not necessary to include any of the machines in the participation, since its use is not an alternative but is obligatory in all the production processes and they are not the direct object of the action. In this case the direct object of the action is butter.

3. **The participation of an element cannot be subject to the participation of other elements.**

If the participation of an element is always produced by the participation of another, its participation should not be contemplated because there is no alternative.

Table 4.3 Elements of Illustration 4.10

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Machines	$i = 1\ldots6$	I_U	Production rate	R_i	C (parts/hour)	W	...
			Modes machines	MM_{ij}	B	S	...
Operators	–	C_D	Quantity	Q	I	W	9
Modes	$j = 1\ldots4$	I_U	Modes machines	MM_{ij}			
Parts	–	C_D	Demand	D	I	W	10.000

Illustration 4.10: Production of Parts

There is a parts production system. For the production of parts, we have 6 machines and 9 operators. A part can be produced in 4 ways:

- Using machine 1 and machine 2
- Using machine 1 and machine 3
- Using machine 4
- Using machine 5 and machine 6

Each machine needs to have an operator for its operation and has a production rate in number of parts per hour. This production rate of parts increases by 20% if two operators are assigned to the machine.

Schedule the production of 10,000 parts in order to minimize the total production time. For simplicity and consistency with respect to the level we are currently at in the methodology, we discard the sequence concept and assume that machine 2 starts working when all the parts assigned to mode 1 have been processed in machine 1 (same with machine 3 and with the production in machines 5 and 6).

Table of Elements

The table of elements can be defined as follows (Table 4.3):

Each machine is individual and unitary because they do not have any measurable properties. MM_{ij} is a binary attribute to identify which machines belong to each part production mode. Operators are a single collective element, since their nine items are identical in the problem and there is no specification on each item individually. Modes that only have binary data must be unitary.

Decision Activities

The action of this system is to produce parts. The parts can be produced using four different forms or production modes. Therefore, each mode represents an alternative production and must participate in the action. However, it would be a mistake to consider the participation of the machines in the action of producing, even though the parts are produced in the machines. Considering the modes of production, each mode requires the participation of a subset of machines, those included in the MM_{ij} attribute, so that the participation of the machine elements is subject to production modes and therefore their participation should not be considered.

Table 4.4 Elements of Illustration 4.11

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Suppliers	$i = 1...N$	I_U	Origin	P_i	B	W	...
T-shirts	$j = 1...10$	C_D	Demand	DT_j	I	W	...
Trousers	$k = 1...3$	C_D	Demand	DP_k	I	W	...

Correct Decision Activity
Action: PRODUCE [parts].
Participating elements: Parts (C_D), Modes ($j = 1...4$ I_U).
Quantification: Integer.
Events: Parts ⇒ Modes $j = 1...4$.
Decision variables: x_j = Number of parts produced in mode j; $j = 1...4$.

Incorrect Decision Activity
Action: PRODUCE [parts].
Participating elements: Parts (C_D), Modes ($j = 1...4$ I_U), Machines ($i = 1...6$ I_U).
Quantification: Integer.
Events: Parts ⇒ Modes $j = 1...4$ ⇒ Machines $i = 1...6/MM_{ij} = 1$.
Decision variables:
x_j = Number of parts produced in mode j with m.achine i; $j = 1...4$.
$i = 1...6/MM_{ij} = 1$.

4. **Participation of a subset of elements within a set.**

From a table of elements associated into sets, the participation of a set of elements in a decision activity does not have to be complete, but we can only choose a subset of elements whose data meet a certain condition.

This consideration also occurs in the own generation of events, where we are not obliged to contemplate all the options of the combination of elements.

Illustration 4.11
Let us suppose a system of purchasing units of our products from a list of N suppliers. We purchase t-shirts of ten different models and trousers of three models. Demand data for each model is known. We have an attribute for the supplier regarding its origin (1: National; 0: Foreign). Foreign suppliers do not supply trousers.

Table of Elements

The table of elements for this description has the following structure (Table 4.4):

Since the text points to the number of t-shirts and trousers demanded, we class these elements as collectives.

Decision Activities

If we establish “Buy T-shirts and Trousers” as a decision activity, but taking into account that trousers are only purchased from national suppliers, the definition of the activity should reflect the options allowed:

Action: PURCHASE [T-shirts AND Trousers FROM Suppliers].
Participating elements: T-shirts ($j = 1\ldots10$ C_D); Trousers ($k = 1\ldots3$ C_D); Suppliers ($i = 1\ldots N$ I_U).
Quantification: Integer.
Events:
T-shirts $j = 1\ldots10 \Rightarrow$ Suppliers $i = 1\ldots N$.
Trousers $k = 1\ldots3 \Rightarrow$ Suppliers $i/P_i = 1$.
Decision variables:
x_{ij} = Number of t-shirts of model j purchased from supplier i; $i = 1\ldots N, j = 1\ldots10$.
y_{ik} = Number of trousers of model j purchased from supplier i; $i/P_i = 1$, $k = 1\ldots3$.

In the same way we could have considered two decision activities regarding the action “Purchase”:

Decision Activity 1:
Action: PURCHASE [T-shirts FROM Suppliers].
Participating elements: T-shirts ($j = 1\ldots10$ C_D); Suppliers ($i = 1\ldots N$ I_U).
Quantification: Integer.
Events: T-shirts $j = 1\ldots10 \Rightarrow$ Suppliers $i = 1\ldots N$.
Decision variables:
x_{ij} = Number of t-shirts of model j purchased from supplier i; $i = 1\ldots N, j = 1\ldots10$.

Decision Activity 2:
Action: PURCHASE [Trousers FROM Suppliers].
Participating elements: Trousers ($k = 1\ldots3$ C_D); Suppliers ($i = 1\ldots N$ I_U).
Quantification: Integer.
Events: Trousers $k = 1\ldots3 \Rightarrow$ Suppliers $i/P_i = 1$.
Decision variables:
y_{ik} = Number of trousers of model j purchased from supplier i; $i/P_i = 1$, $k = 1\ldots3$.

5. **Elements with participation subject to conditions.**

When the participation of an element in a decision activity is subject to that element, it fulfills some condition on other variables (decision activities or calculations), something that a priori is not determinable, and its participation in the activity will always be considered, and the condition of the definition in the activity will be excluded.

Subsequently, a specification will control the values of that activity depending on the condition. Let us take a look at an example.

Table 4.5 Elements of Illustration 4.12

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Centers	$i = 1\ldots3$	I_U					
Vehicles	$j = 1\ldots10$	I_U					
Pallets		C_I					

Illustration 4.12
There is a system in which vehicles are allocated to delivery centers. There are ten vehicles and three delivery centers, each with a given location. For vehicles that are assigned to more than one center, we also have to decide on the number of pallets assigned.

Table of Elements

Although the text is simple and does not incorporate a lot of information regarding data and specifications, with the information that is provided on each set of elements (centers, vehicles, and pallets), we can define centers and vehicles as individual elements and pallets as collective. It specifies that each center has own values in its data. Regarding the vehicles, they are identical but are defined as unitary by the specification stating that "For vehicles that are assigned to more than one center, we also have to decide on the number of pallets assigned," where each vehicle is considered individually (we could have replaced "vehicles" for "each vehicle"). Regarding pallets, we decided to consider them collectives because they are identical items with an indeterminate number in the system.

We define therefore the following table of elements (Table 4.5).

Decision Activities

From the text, it is easily perceived that there are two activities: assign vehicles to centers and assign pallets to vehicles. In this second activity, it is not a priori determined which vehicles will participate, since it depends on the activity of assigning centers to vehicles, since the statement states that "For vehicles that are assigned to more than one center, we also have to decide the number of pallets assigned." As we have said, in these cases it is necessary to consider the participation of all vehicles, without any condition. In the specifications section, it will be necessary to contemplate this condition. Therefore, the activities would be defined as follows:

Decision Activity 1:
Action: ASSIGN [Centers to Vehicles].
Participating elements: Centers ($i = 1\ldots3$ I_U); Vehicles ($j = 1\ldots10$ I_U).
Quantification: Binary.
Events: $i = 1\ldots3 \Rightarrow j = 1\ldots10$.
Decision variables: $\alpha_{ij} = 1$ if we assign center i to vehicle j; 0 otherwise. $i = 1\ldots3$. $j = 1\ldots10$.

Decision Activity 2:
Action: ASSIGN [Pallets to vehicles].
Participating elements: Vehicles ($j = 1\ldots10$ I_U); Pallets $C_{I.}$
Quantification: Integer.
Events: Pallets $\Rightarrow j = 1\ldots10$.
Decision variables:
x_j = Number of pallets assigned to vehicle j; $j = 1\ldots10$.

4.4 Quantification of the Activity

Every system requires an analysis of its decision activities, an analysis to determine what I need to obtain from the decisions. Among the elements participating in the action, the element that acts as direct object to the action, its capacity to be measurable and the quantitative analysis of the action on the element, must determine the type of variable that is generated in the decision activity.

The direct object is a grammatical issue necessary to understand the quantification of the activity. The direct object is the recipient of the action; it is the thing being acted upon, the receiver of the action. Let us see some examples:

- "Purchase products from suppliers": The products are the direct object of the action, what I purchase. The suppliers are an indirect object.
- "Making butter at the factory": Butter is the direct object of the action, what I make.
- "Placing the object on the shelf": The object is the direct object of the action.

When an action is measurable, the element being measured must necessarily act as direct object. Carrying out a previous analysis of the elements that can be measured helps to establish the quantification of the activity. Let us look at some illustrations on quantifying decision activities.

Illustration 4.13
There is a set of 10 workers and a set of 25 jobs. Each job has an affinity value between 0 and 1 with each operator. It involves assigning jobs to operators so that each operator does at least 2 jobs and maximizes the total affinity of the assignment. Each work must be assigned to a single operator.

Table of Elements

The elements that the description reflects are the jobs and the workers. Affinity is a shared attribute between each job and each worker.

The activity of the system "assign jobs to workers" does not measure either workers or jobs but has a logical meaning or choice between elements.

Each job is unitary for several reasons:

Table 4.6 Elements of Illustration 4.13

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Jobs	$i = 1\ldots25$	I_U	Affinity	A_{ij}	C	S	. . .
Workers	$j = 1\ldots10$	I_U		A_{ij}			

– Each job is individual because it has an attribute, affinity on each worker, with an own value not necessarily equal to the rest.
– It is also individual because it refers in an individual way to each job in the specifications (each job must be assigned to a single operator).
– Each job does not have the capacity to be divisible in the system, so it cannot be measured continuously, and therefore will have a unitary character.

The same attribute of affinity also makes us consider the workers as individual elements, and since their data have different values, they are differentiated. Also, the specifications treat each operator in a particular way. On the other hand, workers do not have any property with the capacity to be measurable, so they are also unitary elements.

The table of elements corresponds to the following structure (Table 4.6).

Decision Activities

As we have said, the only action in the system is "assign jobs to workers."

Action: ASSIGN [Jobs to Workers].
Participating Elements: Jobs $i = 1\ldots25$ I_U; Workers $j = 1\ldots10$ I_U.
Quantification: Binary.
Events: $i = 1\ldots25 \Rightarrow j = 1\ldots10$.
Decision variables:
$\alpha_{ij} = 1$ if I assign Job i to Worker j; 0 otherwise. $i = 1\ldots25, j = 1\ldots10$.

The works are the element that I assign to the operators, which act as an indirect object of the action.

In the example a significant characteristic is revealed, the same action can be expressed using another element as a direct object. We could also have written: Assign Operators to Jobs. In that case, the quantification would not have changed because the workers are also unitary individual elements.

Illustration 4.14
We add the following information to the problem of Butter Production:

The system must also assign three workers from the factory to the production of each type of butter. The company has a staff of 15 workers, and there is a cost for the allocation of each worker to each type of butter.

Table of Elements

Table 4.7 Elements of Illustration 4.14

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Machines	$i = 1\ldots2$	I_U	Usage time	T_i	C (Min)	W	. . .
			Time consumed by 1 kg of butter j in machine i	TM_{ij}	C (Min)	S	. . .
Butters	$j = 1\ldots2$	I_M	Profit	B_j	C ($)	W	. . .
				TM_{ij}			
			Worker cost	C_{kj}	C ($)	S	. . .
			Number of workers	N_j	I	W	3
Workers	$k = 1\ldots15$	I_U		C_{kj}			. . .

The new functions of the system add information to the table, which now has the following structure (Table 4.7).

The workers are incorporated as unitary elements, sharing with each type of butter the cost of assigning each worker to each type of butter.

In addition, each butter also incorporates the number of workers needed, 3, the same amount for each butter. Although the attribute refers to the number of workers, the attribute is own because it cannot be shared with each worker, because each worker can assume only its individual information. If workers were a collective element, it would also be the owner of the attribute.

Decision Activities

"The system also has to assign three workers from the factory to the production of each type of butter."

In addition to producing butter, the system presents a new decision activity, "Assign workers to each type of butter."

As we discussed, no numerical data is included in the definition of an activity, so we ignore the concept of assigning "three" workers to each butter. This information will be used in the specifications. The activity would be configured as follows:

Action: ASSIGN [Workers to Butters].
Participating elements: Workers $k = 1\ldots15$; Butters $j = 1,2$.
Quantification: The action falls on the workers, so it is a logical activity.
Events: Workers $k = 1\ldots15 \Rightarrow$ Butters $j = 1,2$.
Decision variables: $\alpha_{kj} = 1$ if I assign butter j to *worker* k; 0 otherwise.

If we try to define the activity as "Assign butters to workers," the butters as direct objects, which are measurable elements, we must realize that the amount we assign to a worker will correspond to the total butter produced or none; therefore we are not measuring the butters, we are defining a logical decision.

To illustrate measurable activities, we propose a classic problem in the world of mathematical optimization and another example in which most of the participating elements are continuous and measurable.

Illustration 4.15

A company has m warehouses where its products are located. Each warehouse A_i ($i = 1 \ldots m$) has a stock of K_i units. There is a set of n Customers ($j = 1 \ldots n$) with a demand D_j of product units. The company must supply the customers' demand of products from the warehouses. The cost of sending a product from each warehouse A_i ($i = 1 \ldots m$) to each customer C_j ($j = 1 \ldots n$) is estimated in c_{ij}.

Table of Elements

The elements that are identified are:

– The company: the system itself, an implicit individual element in all problems.
– The products: element formed by a set of identical items. It will be considered collective since we do not need to consider each product unit individually.
– The m warehouses (A_i, $i = 1 \ldots m$): Each store is necessary to consider it individually since it has an attribute with own value, the number of products. The use of this attribute will be measured in the system, although this is already included in the quantitative nature of the product, so we can consider each warehouse as unitary. When elements are defined abstractly with an index, they are already being defined in a set.
– The n customers (C_j, $j = 1 \ldots n$): Same as the warehouses, it is a set of unitary elements.

Nouns that refer to data:

– Stocks, K_i product units that each warehouse owns.
– The demand D_j of product units that each client owns.
– The cost of sending a product from each warehouse to each customer.

All data that refer to product units are also attributable to the product and therefore are shared with it.

All this is reflected in the following table of elements (Table 4.8).

Decision Activities
Action: SEND [products from warehouses to customers].
Participating elements:
Products (What do I send?) *Direct object.*
Warehouses (Where do I send it from?).

Table 4.8 Elements of Illustration 4.15

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Warehouses	$i = 1 \ldots m$	I_U	Stock	K_i	I	S	…
			Cost	C_{ij}	C	S	…
Customers	$j = 1 \ldots n$	I_U	Demand	D_j	I	S	…
				C_{ij}			
Products	–	C_D		K_i; D_j; C_{ij}			

Customers (Who do I send it to?).
Quantification: Integer.
Events: Product $\Rightarrow i = 1...m \Rightarrow j = 1...n$.
Decision variables:
x_{ij} = Number of product units sent from warehouse *i* to customer *j*.

The text names another action, *supply*. This action can be considered in the text as equivalent to *send*, assuming the same participating elements. If instead we define it as an activity in which only each client and the product participate, we would make the mistake of using an action with determined value as a decision activity. The quantity of products to supply to each customer j is a known value, its demand D_j.

Illustration 4.16
To make two mixtures, M1 and M2, it is necessary to mix four compounds A, B, C, and D. Of the compounds A, B, and C, we need between 20% and 40% of the same in the mixtures. If the content of compound A is higher than compound B in the mixture M1, it is necessary to introduce compound D in an amount equal to 5% of A.

The costs per kilo of A, B, C, and D are, respectively, CA, CB, CC, and CD.

Determine the composition of the most economical mixtures if I must make a total of 25 kg of mixtures.

Table of Elements

The elements that are identified as actors in the problem are the two mixtures and the four compounds. Mixtures must be made, an undetermined amount, by mixing compounds. Since the mixtures have an undetermined quantity, of the compounds, we are also going to use an undetermined quantity, which defines them as measurable. The amount of mixing will be obtained by a function of the compounds, their sum. Therefore, mixtures will not be measured in decisions but in specifications.

The table of elements collects all the information (Table 4.9).

In order to unify sets, the minimum and maximum data has been established for the four compounds, with the following values (Table 4.10).

Decision Activities

A priori, they are identified as actions in the system, *make mixtures and mix compounds in mixtures*. When we refer to introducing compound D, we are referring

Table 4.9 Elements of Illustration 4.16

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Mixtures	$i = 1...2$	I_U	Minimum	N_{ij}	C (%)	S	...
			Maximum	M_{ij}	C (%)	S	...
Compounds	$j = 1...4$	I_M	Cost	C_j	C ($/kilo)	W	...
				N_{ij}			
				M_{ij}			
System	–	I_U	Total mixtures	T	C (kg)	W	25

Table 4.10 Minimum and maximum values

	M1		M2	
	Minimum	Maximum	Minimum	Maximum
A	20%	40%	20%	40%
B	20%	40%	20%	40%
C	20%	40%	20%	40%
D	0%	100%	0%	100%

to the very action of mixing compound D. Regarding the activity of making mixtures, it is not really a decision activity but an action with calculated linear value. The amount of a mixture made is the sum of the compounds that compose it.

Mixing compounds in mixtures: The direct objects are the compounds that will be the elements that are measured in each event.

Action: MIX [compounds in mixtures].
Participating elements:
Compounds $j = 1\ldots4$.
Mixtures $i = 1{,}2$.
Quantification: Continuous.
Events: $j = 1\ldots4 \Rightarrow i = 1{,}2$.
Decision variables:
x_{ij} = Amount (Kgs) of compound j that are mixed in the mixture i.

4.5 Union of Activities

In some systems, there is the possibility of joining activities that are closely related to each other. It happens when there is a logical activity, which we will call secondary, with some elements that also participate in another (logical or measurable) activity, which we will call primary. The two activities can be combined in a single activity that incorporates the options for choosing the secondary activity to the primary activity, provided that there is a conditional relationship between them. The union is not valid in all cases. The relationship that must exist between both so that the union of activities can be carried out is the following:

x = primary activity — *If* $y > 0$ *then* $x = y$ *and* $\alpha = 1$
α = secondary logical activity — *If* $y = 0$ *then* $x = 0$ *or* $\alpha = 0$
y = union activity

In the union activity, the action of the primary activity is maintained, assimilating the secondary activity into the definition itself.

It must be said that in the majority of cases, the union processes are inefficient since they multiply the number of decision variables of the problem. Only in cases where they reduce specifications can they make any sense. There may be cases in which the union embeds specifications between primary and secondary activity. In

Table 4.11 Elements of Illustration 4.17

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Supermarkets	$i = 1...45$	I_U					
Distributors	$j = 1...8$	I_U					
Vehicles	$k = 1...6$	I_U					

general, I discourage this option in the modelling of problems, although it is necessary to incorporate it into the methodology as something that we can find in the formulation of models.

Let's see some illustrations of a union process.

Illustration 4.17
There is a set of supermarkets (i = 1...45), a set of distributors (j = 1...8), and a set of vehicles (k = 1...6) of distribution. The system must assign distributors to vehicles and also assign distributors to supermarkets.

Although data have been omitted, we are going to consider each element as different from the rest and therefore individual and unitary.

Table of Elements (Table 4.11).

Decision Activities

Two activities are identified, on the one hand, to decide the assignment of distributors to supermarkets and, on the other hand, to assign vehicles to distributors.

Decision Activity 1:
Action: Assign [Supermarkets to distributors].
Participating elements: Supermarkets ($i = 1...45$ I_U); Distributors ($j = 1...8$ I_U).
Quantification: Binary.
Events: $i = 1...45 \Rightarrow j = 1...8$.
Decision variables:
$\alpha_{ij} = 1$ if I assign Distributor j to supermarket i; 0 otherwise.
$i = 1...45, j = 1...8$.

Decision Activity 2:
Action: Assign [distributors to vehicles].
Participating elements: Distributors ($j = 1...8$); Vehicles ($k = 1...6$).
Quantification: Binary.
Events: $i = 1...45 \Rightarrow j = 1...8$.
Decision variables:
$\beta_{ij} = 1$ if I assign Distributor j to Vehicle k; 0 otherwise. $j = 1...8$, $k = 1...6$.

Union of Activities
Action: Assign [supermarkets to distributors with vehicles].

Table 4.12 Number of variables in Illustration 4.17

Setup 1	Setup 2: Union
Activity 1: 45*8 = 360 variables Activity 2: 8*6 = 48 variables Total = 408 variables	Activity: 45*8*6 = 2160 variables

Table 4.13 Elements of Illustration 4.18

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Clients	$i = 1\ldots10$	I_U	Route_Client	RC_{ik}	B	S	...
			Demand	D_i	C	S	...
Vehicles	$j = 1\ldots5$	I_U					
Routes	$k = 1\ldots7$	I_U		RC_{ik}			
Merchandise	–	I_M		D_i			

Participating elements: Supermarkets ($i = 1\ldots45$); Distributors ($j = 1\ldots8$). Vehicles ($k = 1\ldots6$).
Quantification: Binary
Events: $i = 1\ldots45 \Rightarrow j = 1\ldots8 \Rightarrow k = 1\ldots6$.
Decision variables:
$\alpha_{ijk} = 1$ if I assign Distributor j "with" Vehicle k to Supermarket i; 0 otherwise. $i = 1\ldots45, j = 1\ldots8; k = 1\ldots6$.

In the variable definition, the second assignment activity is embedded in the "with" preposition. If we analyze the number of variables generated with each configuration, we will check the low viability of the union (Table 4.12).

Illustration 4.18

There is a distribution merchandise system. There is a set of clients (i = 1 ... 10) with a merchandise demand D_i, a set of vehicles (j = 1 ... 5) and a set of routes (k = 1 ... 7). Each route passes through a subset of clients that are collected in the RC_{ik} *attribute:*

RC_{ik} = *1 if Route k passes through Client i; 0 if not.*

Vehicles must select delivery routes and from these routes serve merchandise to customers.

We do not add more elements to the problem, since for the illustration this description suffices. Logically, you could enter trips, days, and also impose specifications of capacity, time, etc.

Table of Elements (Table 4.13)

Since no information about the merchandise is specified, we assume it is continuous.

Decision Activities

Two activities are identified: select distribution routes to vehicles and serve merchandise to customers. Let us see how they are configured:

Decision Activity 1:
Action: Select [Routes for Vehicles].
Participating elements: Vehicles ($j = 1\ldots5$ I_U); Routes ($k = 1\ldots7$ I_U).
Quantification: Binary.
Events: $j = 1\ldots5 \Rightarrow k = 1\ldots7$.
Decision variables:
$\alpha_{jk} = 1$ if we select Route k for Vehicle j; 0 otherwise. $j = 1\ldots5$, $k = 1\ldots7$.

Decision Activity 2:
Action: Serve [merchandise to customers with vehicles].
Participating elements*: Merchandise (I_M); Clients ($i = 1\ldots10$ I_U); Vehicles ($j = 1\ldots5$ I_U).
Quantification: Continuous measurable.
Events: Merchandise $\Rightarrow i = 1\ldots10 \Rightarrow j = 1\ldots5$.
Decision variables:
x_{ij} = Amount of merchandise served to customer i with the vehicle j.

***:** Obtaining the elements in this action may have some complexity. If we had only placed Merchandise and Clients, we would be defining an action with a determined value, the value of its Demand. If with Merchandise we attend to the question "What do we serve?", with the clients "Whom do we serve?", then with the vehicles we answer the question "With what do we serve?"

Union of Activities
Primary activity: Serve.
Secondary activity: Select.
Action: Serve [merchandise to customers with vehicles by routes].
Participating Elements: Merchandise (I_M); Clients ($i = 1\ldots10$ I_U); Vehicles ($j = 1\ldots5$ I_U); Routes ($k = 1\ldots7$ I_U).
Quantification: Continuous measurable.
Events: Merchandise $\Rightarrow i = 1\ldots10 \Rightarrow j = 1\ldots5 \Rightarrow k/ RC_{ik} = 1$.
Decision variables:
x_{ijk} = Amount of merchandise served to customer i with vehicle j by route k.
The secondary activity of Select is replaced with the preposition "by."

In the generation of events, not all routes must be specified. For each client, only the routes that pass through it must be considered, information collected in the RC_{ik} parameter. This fact, which is not determined by any decision, since it is due to information about the problem, is not controlled in the non-union version of the activities. In that version it would be necessary to include a specification that would control that a vehicle can only serve a customer if it has chosen a route that passes through said client. Let us consider how that specification would be:

Taking the sets of variables.
$\alpha_{jk} = 1$ if we select Route k for Vehicle j; 0 otherwise. $j = 1\ldots5$, $k = 1\ldots7$.
x_{ij} = Amount of merchandise served to customer i with the vehicle j.

We must assume that the quantity of merchandise served to a customer i by a vehicle j must be 0 if that vehicle has chosen a route k that does not pass through customer i ($\mathrm{RC}_{ik} = 0$). Mathematically, it would correspond with the following logical proposition:

$$\forall i, \forall j, \forall k / \mathrm{RC}_{\mathrm{ik}} = 0 : \text{If } \ \alpha_{jk} = 1 \ \text{ Then } \ x_{ij} = 0$$

This specification is not necessary in the union configuration, since the specification is included in the definition of events. We can say that this is an advantage of the union configuration.

4.6 Examples

We are going to obtain the decision activities of the examples presented in examples Sect. 3.9 from the previous chapter. Therefore, we present in each example the description and the table or tables of elements already obtained in that chapter.

4.6.1 Fire Stations (Example 3.9.1; Source: Larrañeta et al. 2003)

An initial study is planned to install two fire stations in an urban area that currently has none. The approach has been adopted to divide the urban area into five sectors and carry out a preliminary analysis of the repercussions of the possible location of the stations in each of the sectors. The average time, in minutes, of answering a call from a fire station located in a certain sector i for an incidence received from each of the sectors j has been estimated in t_{ij}. The average number of calls per day that will occur from each of the five sectors (F_j) has also been estimated. All these values are

Table 1 Frequencies and time between sectors

t_{ij}	Sector 1	Sector 2	Sector 3	Sector 4	Sector 5
Sector 1	5	12	30	20	15
Sector 2	20	4	15	10	25
Sector 3	15	20	6	15	12
Sector 4	25	15	25	4	10
Sector 5	10	25	15	12	5
Frequency	2,5	1,6	2,9	1,8	3,1

Table 4.14 Version 4.1 of the Elements in Example 4.6.1

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Sectors	$i,j = 1\ldots5$	I_U	Time	t_{ij}	C	S	. . .
			Frequency	f_i	C	W	. . .
Fire stations	$k = 1,2$	I_U					

Table 4.15 Version 4.2 of the Elements in Example 4.6.1

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Sectors	$i,j = 1\ldots5$	I_U	Time	t_{ij}	C	S	. . .
			Frequency	f_i	C	W	. . .
Fire stations	–	C_D	N° items	N	I	W	2

shown in Table 1. *For example, it takes 12 minutes to go from a station located in sector 3 to an incident from sector 5. The last row shows the average daily frequency of calls made to the fire service.*

Version 4.1
Table of Elements (Table 4.14)

Decision Activities
Action: *Install [fire stations in sectors].*
Participating Elements: Fire stations ($k = 1,2$ I_U); Sectors ($i = 1\ldots5$ I_U).
Quantification: Binary.
Events: $k = 1,2 \Rightarrow i = 1\ldots5$.
Decision variables: $\alpha_{ki} = 1$ if we install fire station k in sector i; 0 otherwise.

Version 4.2
Table of Elements (Table 4.15)

Decision Activities

Action: Install [fire stations in sectors].
Participating elements: Fire stations C_D; Sector ($i = 1\ldots5$ I_U).
Quantification: Integer.
Events: Fire stations $\Rightarrow i = 1\ldots5$.
Decision variables: x_i = Number of fire stations installed in sector i.

Table 4.16 Table of elements in Example 4.6.2

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Banquets	$i = 1\ldots4$	I_U	Tablecloths	m_i	I	S	...
			Day	d_{it}	B	S	...
Storeroom	–	I_U	Stock	S	I	S	200
Market	–	I_U	Price	p	C	S	12
Basket	–	I_U					
Laundry	–	I_U					
Fast wash	–	IU	Cost	cF	C	S	6
Slow wash	–	IU	Cost	cL	C	S	4
Days	$t = 1\ldots4$	IU		dit			
Tablecloths	–	CI		mi; S; p; cF; cL			

4.6.2 *Food Service (Example 3.9.2; Source: Larrañeta et al. 2003)*

A food service business has contracted four banquets for the next 4 days, requiring 150 clean tablecloths for the first banquet, 100 for the second, 140 for the third, and 130 for the fourth. Currently, it has 200 tablecloths in the storeroom, all of them clean, and they can buy what you need on the market every day at a cost of 12 m.u./ tablecloth.

After the banquets, the tablecloths can go to the laundry basket or send them to wash in the laundry. The laundry offers the following washing service:

- *Fast: Clean tablecloths for the next day, at a cost of 6 m.u/tablecloth.*
- *Slow: Clean tablecloths for 2 days, at a cost of 4 m.u/tablecloth.*

Table of Elements (Table 4.16)

Decision Activities
Action: Buy [tablecloths every day in the market].
Participating elements: Tablecloths C_I; Days ($t = 1\ldots4$) I_U; Market I_U.
Quantification: Integer.
Events: Tablecloths ⇒ $t = 1\ldots4$ ⇒ Market.
Decision variables: x_t = Number of tablecloths bought on day t in the market.

Action: Take [tablecloths after the banquets to the basket].
Participating elements: Tablecloths C_I; Banquets ($i = 1\ldots4$) I_U; Basket I_U.
Quantification: Integer.
Events: Tablecloths ⇒ $i = 1\ldots4$ ⇒ Basket.
Decision variables:
y_i = Number of tablecloths taken to the basket after the banquet i; $i = 1\ldots4$.

Action: Send [tablecloths to the laundry in wash modes after the banquets].
Participating elements: Tablecloths C_I; Banquets ($i = 1\ldots4$) I_U; Laundry I_U; Fast clean; Slow clean.
Quantification: Integer.
Events: Tablecloths ⇒ $i = 1\ldots4$ ⇒ Laundry ⇒ Fast wash.
Tablecloths ⇒ $i = 1\ldots4$ ⇒ Laundry ⇒ Slow wash.
Decision variables:
zF_i = Number of tablecloths sent to wash the laundry in fast wash after banquet i; $i = 1\ldots4$.
zS_i = Number of tablecloths sent to wash the laundry in slow wash after banquet i; $i = 1\ldots4$.

In this last action, it was necessary to include the type of washing process, fast or slow, in the participation. If we had only included laundry, we would not have known how to wash the tablecloths, and therefore, we would not know when they are available. Similarly, the participation of the laundry could have been overlooked, as it is implicit.

In the last two actions, we could have swapped the participation of the banquet with the day, since each banquet i corresponds with each day t:

Action: Take [tablecloths on day t to the basket].
Action: Send [tablecloths to the laundry in wash modes on day t].

In the chapter dedicated to specifications, we will see how this type of problem, where a measurable element is subject to activities over a set of periods, can be represented graphically to facilitate the obtaining of activities, auxiliary calculations, and specifications of equilibrium between the different variables associated with the measurable element.

4.6.3 Location of TV Cameras (Example 3.9.3; Source: Larrañeta et al. 2003)

CPL has to televise the game of the year. The producers have identified 10 possible locations for the installation of cameras and 25 stadium areas that need to be covered by the cameras. The table below indicates the relationship between both:

Location	Covered Area
1	1, 3, 4, 6, 7
2	8, 4, 7, 12
3	2, 5, 9, 11, 13
4	1, 2, 18, 19, 21
5	3, 6, 10, 12, 14
6	8, 14, 15, 16, 17
7	18, 21, 24, 25

(continued)

Table 4.17 Table of elements in Example 4.6.3

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Areas	$i = 1\ldots25$	I_U	Coverage	C_{ij}	B	S	
			Minimal coverage	m_i	I	S	
Locations	$j = 1\ldots10$	I_U		C_{ij}			
Cameras	–	C_I		m_i			

Location	Covered Area
8	2, 10, 16, 23
9	1, 6, 11
10	20, 22, 24, 25

Each area of the stadium must be covered by a camera
Location 9 must have a camera
Areas 1 and 2 require coverage of at least two cameras.
The objective is to minimize the number of cameras used (Table 4.17).

Decision Activities
Action: Install [cameras in locations].
Participating Elements: Cameras C_I; Locations ($j = 1\ldots10$ I_U).
Quantification: Integer.
Events: Cameras $\Rightarrow j = 1\ldots10$.
Decision variables: x_j = Number of cameras installed in location j.

In this type of system, the decision variables are defined as binaries:

$x_j = 1$ if we install camera in location j; 0 otherwise.

This way the definition of the maximum number of cameras that we can install in a location is included in the variable definition, because it is understood. However, our methodology would define that specification where it should be defined, in the specifications section, and the variable is defined as integer. In the specifications, the maximum number of cameras is established: $x_j \leq 1$. Of course, it is understood that the variable x_j can be defined as binary.

"Cover" would not be a decision activity in the system as it is a determined value action, we have to provide coverage, it is not an option, and therefore we do not have to decide anything, so it is a specification.

4.6.4 Trip Planning (Example 3.9.4)

There is a system that assigns travellers to buses. We have a group of 180 travellers who have hired the services of the BUSTOUR Company for today. There are five trips offered. Each traveller has chosen one of the five excursions.

Table 4.18 Table of elements of Example 4.6.4

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Travellers	$i = 1 \ldots 180$	I_U	Trip choice	E_{ij}	B	S	...
			Language choice	I_{ik}	B	S	...
Trips	$j = 1 \ldots 5$	I_U		E_{ij}			
Languages	$k = 1,2$	I_U		I_{ik}			
Buses	$r = 1 \ldots 8$	I_U	Capacity	K_r	I	S	60
Seats	–	C_D		K_r			

Each traveller also chooses the language (English and Spanish) for explanations. It has three options of choice:

- *Spanish*
- *English*
- *Both of them (if they speak both English and Spanish)*

The buses have a capacity of 60 people. There are eight buses.

Each bus that is used must be configured with a language and a trip. Since the explanations are given on the bus journey, it is necessary to place each traveller so that the trip and the language of the explanations that are configured on the bus they are travelling on are compatible with its choice.

BUSTOUR wants to use as few buses as possible to cover the trips (Table 4.18).

Decision Activities

Action: Assign [Travellers to buses].
Participating elements: Travellers $i = 1 \ldots 180$ I_U; Buses ($r = 1 \ldots 8$) I_U.
Quantification: Binary.
Events: $i = 1 \ldots 180 \Rightarrow r = 1 \ldots 8$.
Decision Variables: $\alpha_{ir} = 1$ if I assign traveller i to bus r; 0 otherwise.

Action: Configure [Language in buses].
Participating elements: Languages $k = 1,2$ I_U; Buses ($r = 1 \ldots 8$) I_U.
Quantification: Binary.
Events: $k = 1,2 \Rightarrow r = 1 \ldots 8$.
Decision Variables: $\beta_{kr} = 1$ if I configure language k in bus r; 0 otherwise.

Action: Assign [Trips to buses].
Participating elements: Trips $j = 1 \ldots 5$ I_U; Buses ($r = 1 \ldots 8$) I_U.
Quantification: Binary.
Events: $j = 1 \ldots 5 \Rightarrow r = 1 \ldots 8$.
Decision Variables: $\omega_{jr} = 1$ if assign trip j to bus r; 0 otherwise.

This last activity could also be understood as a logical calculation. The travelers that you assign to a bus condition the trip that you assign to the bus.

Table 4.19 Reduced table of elements of Example 4.6.4

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Groups of travellers	$i = 1...15$	C_D	Trip choice	E_{ij}	B	S	...
			Language choice	I_{ik}	B	S	...
Trips	$j = 1...5$	I_U		E_{ij}			
Languages	$k = 1,2$	I_U		I_{ik}			
Buses	$r = 1...8$	I_U	Capacity	K	I	S	60
Seats	–	C_D		K_r			

$$\forall j, r : \omega_{jr} = 1 \quad \text{IF AND ONLY IF} \sum_{i/E_{ij}=1} \alpha_{ir} > 0$$

If we define it as a decision activity, we cannot forget to define those propositions as specifications. We will see that these specifications are typified within the methodology.

Reduced Table of Elements (Table 4.19)

Regarding the reduced table of elements, the decision activities would be as follows:

Decision Activities
Action: Assign [Travellers to buses].
Participating elements: Travellers $i = 1...15$ C_D; Buses ($r = 1...8$) I_U.
Quantification: Integer.
Events: $i = 1...15 \Rightarrow r = 1...8$.
Decision Variables: x_{ir} = Number of travellers from group i assigned to bus r.
The actions of configuring the language and trip to the buses are identical to those presented for the table of elements 4.18.

Table 4.20 Table of elements of Example 3.9.5

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Tasks	$i = 1...n$	I_U	Start time	S_i	C	W	...
			End time	E_i	C	W	...
			Weight	P_i	C	W	...
Machines	$j = 1...m$	I_U					

4.6.5 Fixed Job Scheduling Problem (Example 3.9.5; Kroon et al. 1995; Kolen et al. 2007)

There is a set of n tasks with a given start and end time and a weight. There is also a set of m machines. This set selects tasks to be processed in the machines so that the selected tasks have a maximum weight. A selected task is processed completely on a single machine. A machine cannot perform two tasks overlapped in time.

Table of Elements (Table 4.20).

Decision Activities

The wording talks of selecting tasks to be processed in the machines. Since the elements are unitary, the activities are going to be logics, a selection process, in this case selecting tasks to be processed in machines. The action that defines the activity is to process tasks in machines. This activity will result in a selection of tasks, those that have been processed. Therefore, it is not necessary to consider the activity of selecting tasks in addition to the task of processing tasks in machines. Selecting tasks corresponds to a logical calculation that is obtained from the activity of processing (if you have processed a task in a machine, you have selected that task).

Action: Process [task in machines].
Participating elements: Tasks $i = 1...n$ I_U; Machines $j = 1...m$ I_U.
Quantification: Binary.
Events: $i = 1...n \Rightarrow j = 1...m$.
Decision variables: $\alpha_{ij} = 1$ if we process task i in machine j; 0 otherwise.

4.6.6 Health Centers (Example 3.9.6)

There is a city containing 12 health centers. Due to population changes, it has been decided to reassign health centers to citizens, a total of n. We know each citizen address, and therefore, the system allows us to know the distance from their homes to each health center. Each health center has a capacity that is expressed in the number of patients that can be seen per day. It is estimated that 1% of people go

Table 4.21 Table of elements of Example 4.6.6

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Health centers	$i = 1...12$	I_U	Distance	D_{ij}	C	S	...
			Capacity	K_i	E	W	...
Citizens	$j = 1...n$	I_U		D_{ij}			
City	–	I_U	Attendance	A	C	W	0,01

daily to health centers. The objective is to minimize the sum of the distances between each citizen and the health center assigned.

Table of Elements (Table 4.21)

Decision Activities
Action: Reassign [citizens to health centers].
Participating elements: Citizens $j = 1...n$ I_U; Health Centers $i = 1...12$ I_U;
Quantification: Binary.
Events: $j = 1...n \Rightarrow i = 1...12$.
Decision variables:
$\alpha_{ij} = 1$ if we reassign citizen i to health center j; 0 otherwise.

References

Kolen, A., Lenstra, J. K., Papadimitriou, C., & Spieksma, F. (2007). Interval scheduling: A survey. *Naval Research Logistics, 54*, 530–543.

Kroon, L. G., Salomon, M., & Van Wassenhove, L. N. (1995). Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research, 82*, 190–205.

Larrañeta, J., Onieva, L., Cortés, P., Muñuzuri, J., & Guadix, J. (2003). *Métodos Cuantitativos en Ingeniería de Organización*. Sevilla: Editorial Universidad de Sevilla.