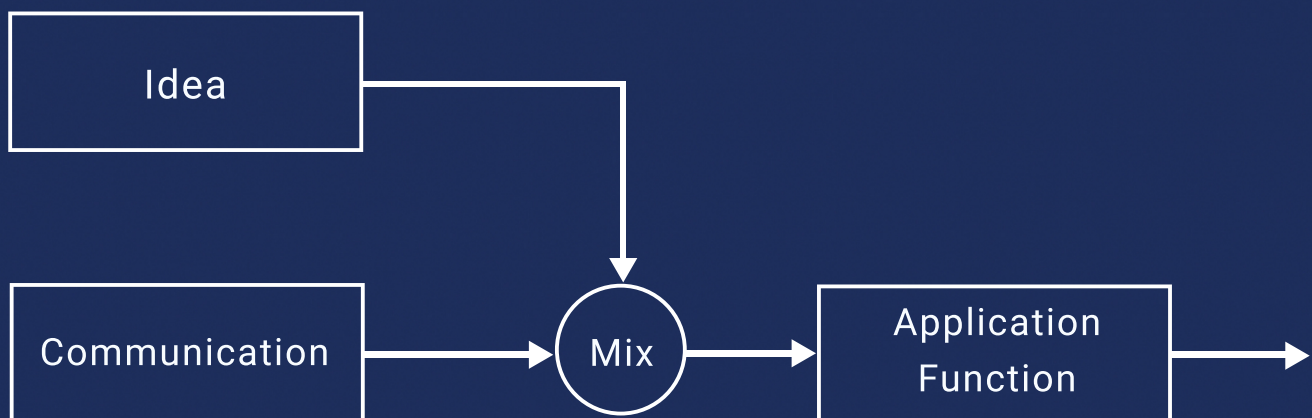


Application Modeling Tutorial & User Manual Communication Domain With Practical Examples Vol. I



The Speak Logic Project

We Promote Better Communication

www.Speaklogic.org

Book #3

*The
Speak Logic Project*

We Promote Better Communication

*Application Modeling Tutorial
and User Manual
Communication Domain*

With

Practical Examples

Vol. I

www.speaklogic.org

Other Resources

Create and Manage Projects

www.trabau.com

Modeling Software

www.slpsoft.com

Speak Logic Forum

www.speaklogic.org/forum

Copyright 2011-2023, first published 2011

Download Software www.slpsoft.com

Books

To help you with the learning of the principle, the following books are given in order. You start from the first chapter—page 1—of *Understanding the Principles of Communication*. If your goal is to learn the principle through an instructor, then there is no need to have books. Otherwise, you should have a total of 10 books in your possession.

In terms of learning the principle, there is no need to be in a hurry. As we live, we need it. In other words, while we are alive, we need the principle. There is no need to be in a hurry while learning it.

Order	Title	ISBN
1	<i>Understanding the Principles of Communication</i> or <i>Identification and Correction of Errors in Communication</i>	9798687646154 or 9798687649179
2	<i>Communication and Information Analysis Guidelines</i> or <i>Application and Project Analysis Guidelines</i>	9798687650250 or 9798687656498
3	<i>Application Modeling Tutorial Communication Domain with Practical Examples Vol. I</i> or <i>Project Modeling Tutorial Communication Domain with Practical Examples Vol. I</i>	9798687667210 or 9798687668422
4	<i>Application Modeling Tutorial Communication Domain with Practical Examples Vol. II</i> or <i>Project Modeling Tutorial Communication Domain with Practical Examples Vol. II</i>	9798687677134 or 9798687678971
5	<i>Fundamental of Communication Vol. I</i>	9798689269771
6	<i>Fundamental of Communication Vol. II</i>	9798689271415
7	<i>Fundamental of Communication Vol. III</i>	9798689272221
8	<i>Application Modeling Tutorial Theory Domain with Practical Examples Vol. I</i> or <i>Project Modeling Tutorial Theory Domain with Practical Examples Vol. I</i>	9798689276182 or 9798689277561

9	<i>Application Modeling Tutorial Theory Domain with Practical Examples Vol. II or Project Modeling Tutorial Theory Domain with Practical Examples Vol. II</i>	9798689279657 or 9798689281636
10	<i>Solution Process Working in Group</i>	9798689283692

*If you are new to modeling, please start
with Appendix A:*

Introduction to Modeling

Table of Contents

Application Modeling **10**

Understanding the Communication Domain	13
Understanding Communication	14

Application Modeling Entity Identification **17**

The Communication Interface Entity	17
The Communication Link Entity	18
The Left Signal and Right Signal Entity	19
The Person Entity	19
The Communication Entity	20
The Word Entity	21
The Sentence and Paragraph Entities	22
The Communication Function Entity	23
The Principles Entity	24
The Application Entity	25
The Communication & Application Mixture Entity	26
The Error Correction Function Entity	28
The Communication Process Entity	29
The Analysis Window Entity	30
The Sentence Analysis Entity	31
The Communication Result Entity	32
The Communication Mixture Entity	33
What We Do Entity	34
The Grouping Entity	35
The Error Entity	36
The Compensator Entity	37
The Problem Entity	38
The Solution Entity	38
The Information Entity	38
The Feedback Entity	39
The Reason Entity	40
The Entity Element	41

The Communication Element Entity	43
The ECF Entity	44
The Definition Entity	45
The Dictionary Entity	46
The Definition Collection Entity	47
The Separation Line Entity	49
The Time Line Entity	50
The Progress Entity	51
The Error in Communication Gives Rise Entity	51
Time Chart and Dateline Entity	55
The Time Entity	56
The Operating Principle Line Entity	56
The Sub Function Entity	56
The Notes Entity	57
The Node Entity	58
The Callout Entity	61
The Node Information Table Entity	61
The Communication Holder Entity	62
The External Communication Holder Entity	63
The Empty Container Entity	64
The Reference Entity	65
The Question Entity	66
The Answer Entity	67
Unknown Entity	68
Number Identification	69
The Information Label	70
The Point to Label	70
The Give Rise Label	71
The Relationship Label	72
The By Label	72
The Dependency Label	73
The Agreement Label	74
The Match Label	74
The Inclusion Label	75
The Attach Label	76
The Interaction Label	76
The Visually Identified	77
Some Communication Related to Work Labels	77
The Use Label	77
Communication Signals Direction	78
Curly Brackets	78

The Relationship Entity	78
Entity Attach to Other Entity	84
Identify Multiple Entities as an Entity	84
Identify Part of an Entity	85
Identify Relationship as an Entity	86
The Location Entity	88
The Mobility Entity	88
The Fixed Entity	89
The Area Entity	90
Entity and Aspect of Entity Model	96
Change of Communication Function from Communication	98
Change of Communication Function from Communication Graphical Form	101
Graph Line and Graph Point	102
Graphical Axis Entity	102
The Continuity Entity	103
Group of People	103
Sub Application and Part of Application	105
Sub Application Result and Part of Application Result	106
Grouping Part of Communication Function	107
Review of Grouping	109
Grouping Communication Holder	109
Grouping Entity and Communication Entity	110
Grouping Error and Compensator	110
Grouping of Problem	110
Grouping of Feedback	111
Grouping of Analysis	111
Grouping of Questions and Answers	111
<i><u>Some Entities Connection Examples</u></i>	<i><u>112</u></i>
<i><u>Conclusion</u></i>	<i><u>166</u></i>
<i><u>Error Identification and Correction Charts</u></i>	<i><u>167</u></i>
<i><u>Exercises</u></i>	<i><u>170</u></i>
<i><u>Appendix A:</u></i>	<i><u>176</u></i>
<i><u>An Introduction to Modeling</u></i>	<i><u>178</u></i>

Representation of Entities	179
<u><i>Flow, Connector, Input, Output, and Interface</i></u>	<u>182</u>
Connection of Entities	183
Flow of Entities	184
Input and Output	185
<u><i>Aspect and Function Representation</i></u>	<u>186</u>
Representing Aspects of Entities	187
Representing Functions of Entities	188
<u><i>Group, Set, List, and Collection Representations</i></u>	<u>189</u>
Using Continuation in Collection	191
Using Indexing	192
Grouping of Entities	192
Set, Subset, and Main Set	193
Grouping of Entities and Types	194
Multiple Inputs and Outputs	195
Feedback and Adjustment	196
More about Grouping and Collection	197
Separation of Entities	200
<u><i>Relationship and Mixture Representation</i></u>	<u>202</u>
Representing Relationship and Mixture	203
<u><i>Complexity in Modeling</i></u>	<u>206</u>
Understanding Complexity in Modeling	208
Steps of Reducing Complexity in Modeling	209
Visual Aspect of Communication in Modeling	210
<u><i>Index</i></u>	<u>212</u>

Application Modeling

Introduction

We connect together through a communication interface that enables us to exchange information. With that communication interface, we can define communication itself as the process of exchanging information. During a communication process, we can exchange information to each other. For example, during an oral communication between you and me, I can repeat a sentence to you; you can also repeat a sentence to me. Whenever we use the term communication here, it means all forms of communication.

Since communication is the only interface that connects us together, everything that we do depends on communication. For instance, we communicate to do our works. With the importance of our communication interface, it is very important for one to understand each other during a communication process. Given that we communicate to satisfy our needs, without the understanding of our communication, our needs would not be satisfied. It is always good to present our communication in the simplest form it can be, so it can be understood. We already know that any error that is developed during the communication process will cause problems in our applications without being corrected. For that reason, it is always good to analyze our communications and correct errors to prevent problems in our applications.

To better understand the process of communication related to what we do, it is always good to separate the communication itself from its purpose. What do mean by that? We mean that it is always good to separate the communication and the purpose of the communication, which is the application. By doing so, we can have a better picture of the application and analyze it during the communication process and correct any error that are presented before its execution. Utilizing that process allows us to correct errors in our communication and provide us with an error free application. In order to do that, the communication itself has to be treated as a separate entity and then analyzed. Given that what we do includes two steps, the communication step and the application execution itself, during the communication step, we can analyze the

communication and make all necessary corrections. In order to have a better view of the communication related to the application, it is always good to have a picture of the overall process. In communication, it is always good to draw the process of the application related to communication to enable us to understand it better. In other words, when we communicate it is always good to draw a picture of what we have in mind or what we intend to do to enable us to understand it better.

To provide us with a better picture of our application related to our communication, it is always good to use diagrams to model them, so we can analyze the overall process. The Speak Logic Application Modeling enables us to draw the process of our applications related to our communications so we can have a better view of our project. By having a picture of our application related to our communication, we can analyze our communication and correct any error that is presented to prevent problems in our application. There are two ways to look at the usage of the Speak Logic Application Modelling. We can use it to model what we do, which is our application. We can also use it to model our organizations. For instance if the function of an organization is to develop a product or provide a service, then the project modeling can be used for that; at the same time, it can also be used to show how that organization organized.

This user manual will provide you with some instructions about using The Speak Logic Application Modeling. Given that the model is not application specific, the aim of this tutorial is not to connect the entities for us, but to provide us with information about each entity, so we can connect them together depends on our applications. Although the Speak Logic Project Modeling provides us with the simplicity to analyze our applications, however it does not carry any weight at all in terms of error analysis and removal. In other words, error analysis and correction from our applications depend on our communication, not on the Speak Logic Project Modeling. The usage of the Speak Logic Project Modeling assumes the principle of communication is understood. While we can use the model to visualize or represent a visual aspect of our application on a computer screen, however the model can also be used without the usage of a computer. For instance, we can use the model on paper or on a drawing board to represent a visual aspect of our application.

It is also important to note that as well, each entity is considered an actual entity of the underlying entity name. In other words, each entity is considered to be the actual representation of the entity name. We use the word entity in this manual to refer to any entity. Whenever we use the word entity with another name to denote an entity, it represents an actual entity of that name. For instance, we can refer to the communication entity as the actual communication.

Whenever we use the word entity here, it means things that exist in life in both physical and nonphysical forms. For instance an entity can be physical, as well as another entity can be nonphysical. It is the same as saying that, physical things and nonphysical things. Here the term nonphysical entity may also refer to entities that may not be visible to everybody. In other words, those entities exist, but some of us may not be aware of them. In this case, some of us think that they don't exist at all. Thus, those entities are not visible to everybody or identified by everybody.

Understanding the Communication Domain

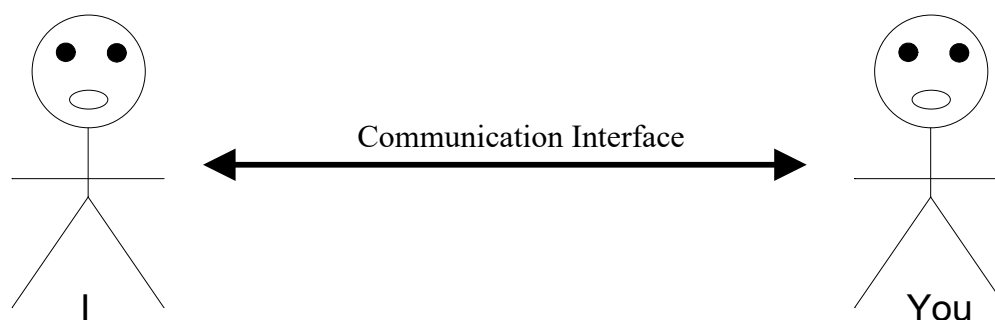
Given that we communicate relatively about entities that we identify; given that the entities that we identify make our communications possible, it makes sense for us to model our application or what we do accordingly to those entities. Since during our application we communicate relatively or accordingly to the function that we are going to execute, it makes sense for us to model that function accordingly to our communication. The communication domain model enables us to do just that. It enables us to model our application according to our communication. The communication domain model enables us to view or portrait our application according to our communication. In other words, the communication domain model enables us to view our application or what we do, as a function of our communication. Here our communication means the communication of the people who are working in the project.

Since our communication ability is one of our aspects; since communication is one of our aspects, in term of what we do, we communicate in order for us to execute our application. In the communication domain, we model our application related to our communication. In other words, in the communication domain, our application or what we do becomes a function of our communication. For instance, if a function of an organization is to provide a service or manufacture a product, then in the communication domain, this function is being viewed as a function of communication. In other words, the service provided by that organization or the product that is manufactured by that organization, is being viewed as a function of communication of the people who work in that organization. It is very important for us to understand the communication domain. The communication domain model enables us to view or model what we do or our function or our application related to our communication.

Understanding Communication

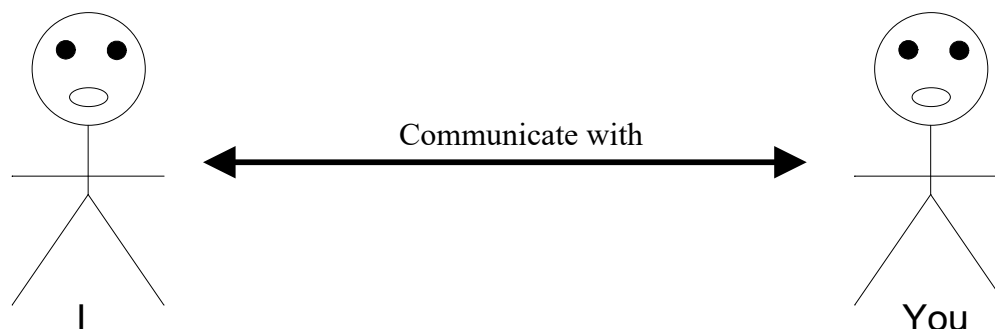
Before we continue, let's look at some important factors of communication. Keep in mind that based on our understanding, there are three signals entities in communication. We understand miscommunication, in other words, we understand when a communication is unsatisfactory in term of its content and its application, we use the red signal entity to denote that understanding. That means we use the red signal entity to denote a communication with error. We understand when a communication is satisfactory, based on its contain and its application. In other words, based on the content of the communication and its function we understand when it is good. We use the green signal entity to denote that understanding. We also understand feedback, in other words, we understand a type of communication that is related to correction or enabled us to make adjustment to what we do. We use the blue signal entity to denote that understanding.

To better understand communication, let's define it as follows. We have already defined communication as the process of sending and receiving information among us. Now, let's define communication relative to the way we have defined it; let's define information as signals, and communication as the process of sending and receiving signals among us. In other words, when we communicate we simply send and receive signals to each other. We do have a sense that enables us to interpret those signals; which is the same as saying that we do have sense that enables us to understand the information that we exchange to each other. To better understand the overall communication process among us, let's look at the model below. The model below shows that we interface together through communication. We can also say that there is a communication interface between us; for instance "I" and "You" interface through communication.

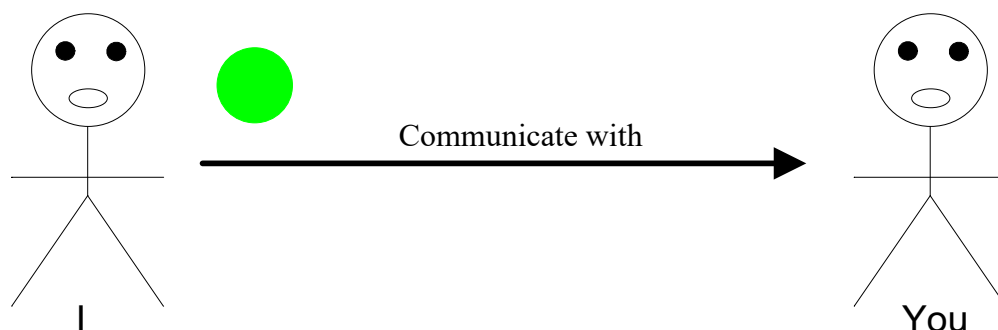


To show the flow of communication, we can replace the communication interface shown above with words that we use in communication. We can also say that rather than using the communication interface phrase on the above model, we can simply replace it by interaction

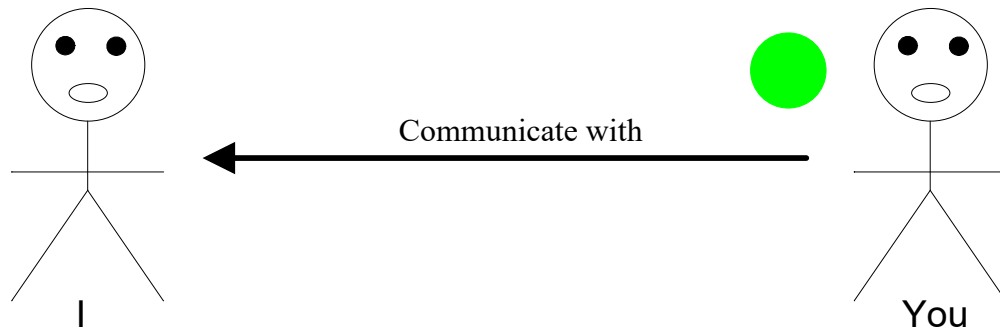
elements or by communication interaction element. The model below shows just that; we simply replace the phrase communication interface with *communicate with*. In this case, the interface shows that *I communicate with you*. We interact with each other through communication. Whenever we use the term interaction elements, it means that communication elements that we use to interact with each other. For example words, phrases, videos, pictures etc. can be viewed as interaction elements. For oral and written communications, words, phrases, drawings, diagrams, models etc. can also be viewed as interaction elements.



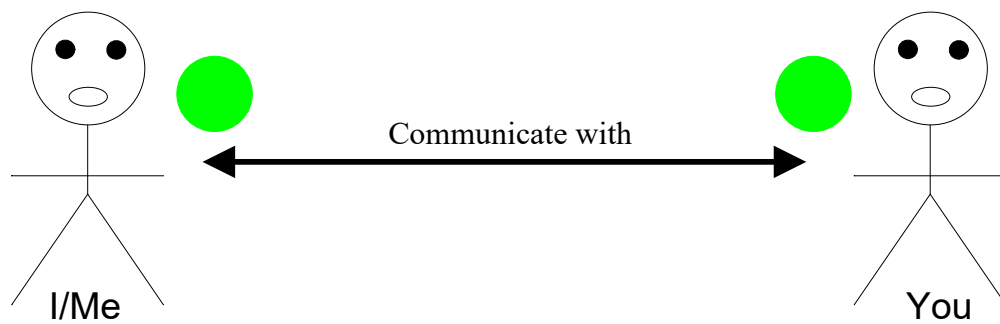
As we defined earlier, communication is the process of sending and receiving signals to each other. To better understand the overall process in terms of flow and diagramming; let's look at the model below. It shows that I communicate with you, which is the same as I send a signal to you.



Given that we do have a sense that enables us to understand or interpret the receiving signals, the model below shows that after you receive the signal that I send you, then you send a signal back to me. From the model above to the one below, it shows that I send you a signal, which is the same as I communicate with you; while the one below shows that you send me a signal which is the same as you communicate with me.



From what we have said above, the overall communication interface which is viewed as the process of sending and receiving signals can be viewed as the model below. The model below is simply a representation of the two models above. It shows a two ways communication between you and me, where I send signals to you and you send signals to me. In other words, I communicate with you and you communicate with me.

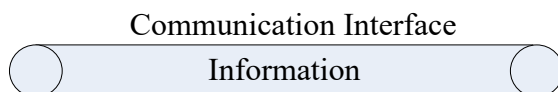


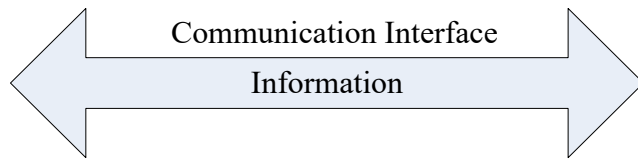
Application Modeling Entity Identification

As we said earlier, we interact with each other through communication and communication enables us to exchange information. We define information that we exchange to each other as communication elements, which are considered signals. The Speak Logic Application Modeling defines three types of signals. Those three types of signals can be considered as tokens. We use the word token here to denote the dots. We could have also used the word dot. As shown below, the three types of signals are the green token, the red token, and the blue token. Each of those signals has a purpose. The table below gives more information about each signal. To better understand the other entities related to the signals, it is good to take it like this; signals flow inside and outside to the other entities.

Signals Type	Explanation
	The green signal entity or the green token signal denotes a communication without error; for example, an input sentence without error
	The red signal entity or the red token signal denotes a communication with error; for example, an input sentence with error
	The blue signal entity or the blue token signal denotes feedback or parent feedback; this signal acts as a compensator to correct errors. For example, if an input sentence contains errors, this signal will remove the errors and compensate for the removal to make sure the output sentence is error free

The Communication Interface Entity





Usage and Description

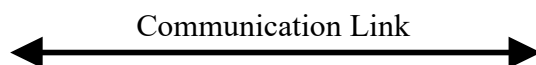
The communication interface path above shows our communication interface with information included. Since information is what flows through our communication interface, this model can be used to show the transmission of information inside our communication interface. In other words, this entity can be used to show that our communication interface carries information.

Available Options

Available options for the communication interface entity include

- Communication interface
- Communication interface path
- Communication interface entity
- Communication link
- Etc.

The Communication Link Entity



Usage and Description

The communication link entity is the same as the communication interface entity shown above. Communication link is simply another name for communication interface. The communication link is used to show signal paths. For example, in a two-way communication between you and me, the communication interface shows the direction of the signals between you and me and between you and me. The communication link entity is used to show the directions of our communications.

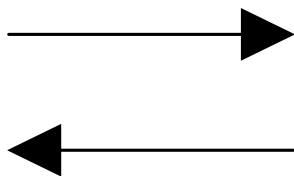
Available Options

Available options for the communication link path include

- Communication interface

- Communication link
- Communication link entity
- Communication link path
- Communication path
- Etc.

The Left Signal and Right Signal Entity



Usage and Description

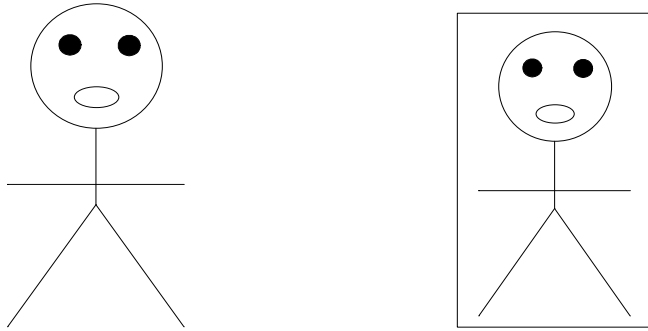
The left signal entity and the right signal entity show the direction of the signal flow or the flow of the communication. For example, if I communicate with you, which is the same as I send a signal to you, the right signal entity can be used to show that. The same as if you communicate with me, which is the same as you send a signal to me, the left signal entity can be used to show the flow of that communication.

Available Options

Available options for the left signal and the right signal entity

- Left/right arrow
- Left/right signals
- Left/right signals path
- Left signal entity
- Right signal entity
- Left communication flow
- Right communication flow
- Etc.

The Person Entity



Usage and Description

The person entity above is used to send communication signals. For example, I communicate with you, this entity represents me. You send a signal to me, this entity represents you. The framed version of the person entity above enables us to connect it with other entities easily. For more information about connecting the entities, see the example section. This entity can be connected with other entities to show the communication of the person. For example, I use communication to solve this problem. In this case, this person entity can be connected with the communication entity.

Available Options

All available options for the person entity include

- I, Me, You, We, He/She, Him/Her
- Person Name
- Person Title etc.
- Person
- Person entity
- Human entity
- People
- Human
- Etc. Includes, employee, employee name, employee with number or employee number, customer, inspector, person with number, tester etc.

The Communication Entity



Usage and Description

The communication entity is used to show the presence of communication from what we do and the separation of communication from our application. For instance, we can use the communication entity to show communication as a separate entity from our application. The communication entity can be connected to other entities that are related. For example, the communication entity can be connected to the person entity. It can also be connected to the application entity. For example, we use communication to solve this problem. In this case, the communication entity can be connected to the application entity, which is solving problem. The communication entity accepts both input and output signals. It also accepts feedback signals as well. The table below shows the output of the communication entity related to input signal and feedback.

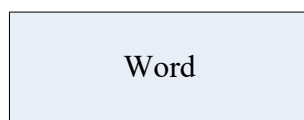
Input	Feedback	Output
Red	None	Red
Green	None	Green
Red	Blue	Green

Available Options

The followings are considered available options for the communication entity. Those options can be replaced by anything that is related.

- Sentence
- Any communication entity
- Any communication
- Communication Entity
- Communication
- Paragraph
- Words
- Etc.

The Word Entity



Usage and Description

The word entity is used to identify a word in our communication. This entity has the same option as the communication entity. As the description says, the word entity can be used to identify a word. For example, the word entity can be used to identify a word in a sentence. The word entity can take both input and feedback. For example, sentence analysis can be done in a word to determine if the usage of that word is understood. In this case, the word entity can be used with the sentence analysis entity, where the sentence analysis entity is used as feedback for the word entity. When that happens, the output of the word entity can be determined by the feedback and the input; see the table below for output result related to input and feedback for the word entity.

Available Options

Available options for the word entity include

- Word
- Sentence
- Paragraph
- Etc.

Input	Feedback	Output
Red	None	Red
Green	None	Green
Red	Blue	Green

The Sentence and Paragraph Entities



Usage and Description

The sentence and the paragraph entities are similar to the word entity. The sentence entity is used to identify a sentence in our communication, while the paragraph entity is used to identify a paragraph in our communication. These two entities have the same options as the word entity. For more information about using the sentence entity and the paragraph entity, refer to the word entity.

Available Options

Available options for the sentence and paragraph entities include

- Word
- Sentence
- Paragraph
- Etc

The Communication Function Entity

Communication
Function

Usage and Description

The communication function entity is considered to be the function of the communication. It is also considered to be the purpose of the communication. We can also call it application function or application execution function, which is viewed as the function of the application. To the right above is just another way of representing the communication function entity. The communication function takes both input and output. We can say that the communication function takes two inputs and one output. Given that our application includes two instances, the communication, and the application itself. The communication function is considered to be a mixture of the communication and the application in this case. For example, if we talk about fixing a car, then fixing a car is considered to be the communication function. We can also say fixing a car is the purpose of the communication. The communication function takes both the communication and the application as inputs. In this case, for some instances we can say that it takes the communication and the picture of the application as inputs. For instance, if we talk about fixing a car, the picture of the application which is the car is considered as the application and what we talk about is considered as the communication. We can also say the car itself is considered to be the application, while what we talk about is considered to be the communication.

Available Options

Available options for the communication function entity include

- Communication function
- Application execution function

- Application function
- Communication purpose
- Communication Reason
- Project Function
- Function of our Project
- Function of what we do
- Etc.

Input	Output
Green	Green
Red	Red

The Principles Entity



Usage and Description

The principles entity is considered to be feedbacks that feed other entities. In this case, we can say the principles entity provides feedbacks to other entities that can accept feedbacks. The principles entity provides parent's feedbacks to other entities by sending a blue signal to entities that accept feedback. The table below shows the output of the other entities when the principles entity provides feedback to them. The principles entity only provide a feedback signal to another entity, when the input of that entity is negative or red; see the table below for output result related to the input of the other entity. The way to look at it, the principles entity always makes sure the output of the entity it is attached to is positive. When it is connected preceding an application, we can say that the principles entity is used to make sure that application is always positive or execute without error.

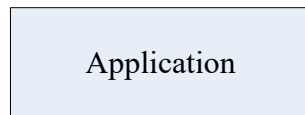
Input	Feedback	Output
Green	None	Green
Red	Blue	Green

Available Options

Available options for the principles entity include

- Principles
- Principle
- Principle of Operation
- Parent principles
- Parent
- Principle Entity
- Parent Principle
- Sentence Analysis
- Error Correction
- Error Analysis
- Analysis
- Error Correction Function (ECF)
- Etc.
-

The Application Entity



Usage and Description

The application entity represents a picture of our application. Depending how we look at it; we can say that the application entity represents the execution of our application. For instance, if we segment our application into two parts, and we say one part is the communication and the other part is the application process. In this case, we can say that our application includes communication and the application itself. If we segment the same application into several parts, where we also include the communication function entity, then we can say that the application entity is a picture of our application. We can also say a picture of the application in our mind.

The usage of the application entity includes the two ways we described above. If we segment our application into two processes, the communication process and the application process itself, in this case the application will depend on the communication. With that, the communication

entity connects to the application entity. From that, we have the table below that shows the output of the application based on the communication entity. Again, it always depends how we look at it. If we use the application entity to show the separation of communication from our application, we can use the application entity as the application execution. However, if we use the application entity to show the step of our application from start to finish, we can say the application entity is simply a picture we have about our application in our mind. Refer to the example section for more information about using the application entity.

Input	Feedback	Output
Green	None	Green
Red	Blue	Green
Red	None	Red

If we segment our application into different sections where we include the communication process or the mixture of communication and the application entity, in this case, the communication entity serves as an input of the communication process or the communication and application mixture.

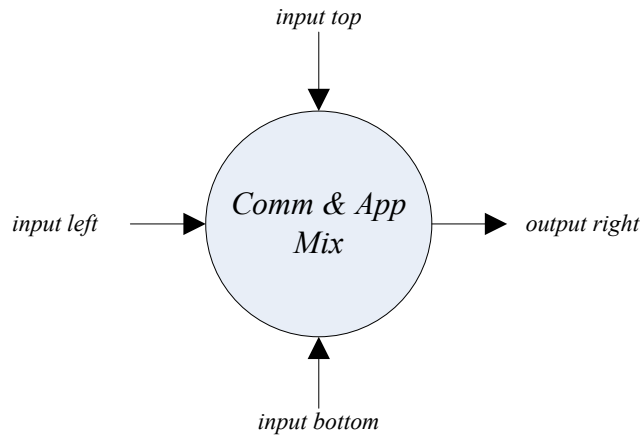
Available Options

Available options for the application entity include:

- Application
- What we do
- Etc.

The Communication & Application Mixture Entity





Usage and Description

The communication and the application mixture entity are used to show the mixture of communication and the application. All of the above representations are the same.

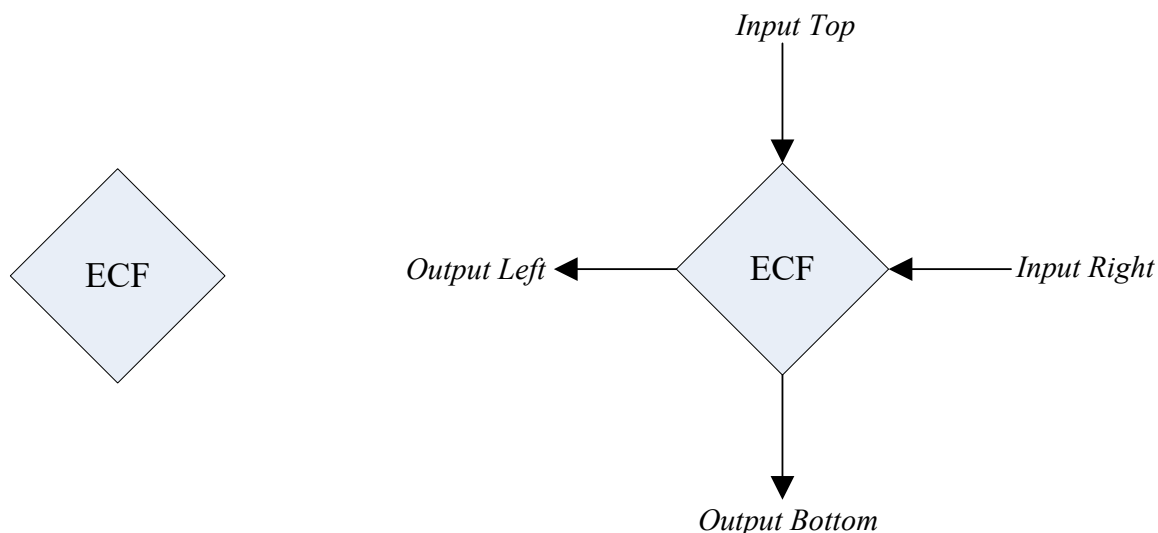
The model above with the labels is the same as the ones without the labels. The table below shows more information about the labels; refer to the example section for more information about connecting the communication and application mixture entity.

Input Left	Input Top	Input Bottom	Output Right
Communications	Application	Feedback	Communication and Application

We use the communication and application mixture entity to show the mixture of the communication and our application after being separated. The communication and the application mixture entity takes three inputs; one input is the communication, one input is the application, and the other input which is optional is the feedback. The output of the communication and application mixture entity is connected directly to the communication function entity. The table below shows the output of the communication application mixture entity, based on the input and feedback. Given that our application result depends on communication, in this case we can say the output of this entity is related to the communication input.

Input Left	Input Bottom	Output Right
Green	None	Green
Red	Blue	Green
Red	None	Red

The Error Correction Function Entity



Usage and Description

This is another version of the error correction function entity that takes two inputs and two outputs. The two inputs are the top one and the right one, while the two outputs are the left one and the bottom one. The Error Correction Function (ECF) entity is used to show a correction process. We can use this entity to remove errors in our communication process for a top down model. The inputs of the ECF related to the outputs can be viewed as follow. The output at the bottom feeds the application, while the one to the left feeds us. The input to the right gets feedback from our parent or the principles, while the one on the top provides the corrected version related to the feedback. The table below shows the inputs outputs result of the ECF.

Input Top	Input Right	Output Left	Output Bottom
Red	Blue	Red	Green

It is always good to look at the ECF entity show above in a timely manner, since the correction is made at the time the principles are given to us and we apply them. Since the error occurs at the time we commit it, it makes sense for us to look at the table in a timely manner. With what we have just said, let's look at the table below from the ECF above by taking time into consideration.

Time 1		Time 2	
Input Top	Input Right	Output Left	Output Bottom
Red	Blue	Green	Green

The table below is the same as the one above. It depends on how we look at it.

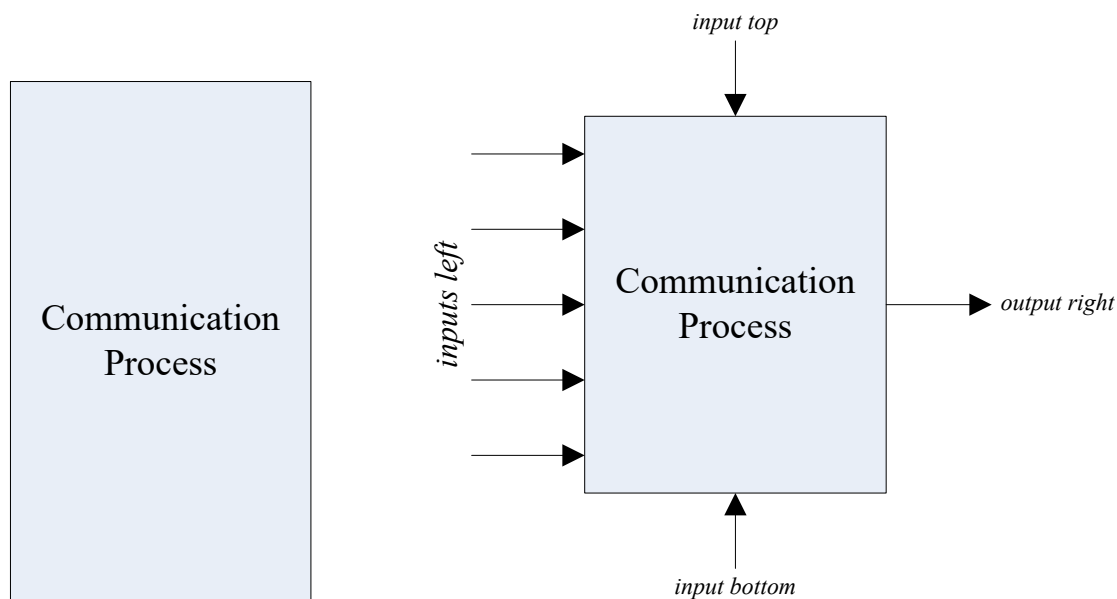
Time 1		Time 2	
Input Top	Input Right	Output Left	Output Bottom
Red	Blue	Blue	Green

Available Options

Available options for the Error Correction Function includes

- ECF
- Error Correction Function
- Error Correction
- Etc.

The Communication Process Entity



Usage and Description

The communication process entity is the process when we communicate about our application. During the communication process, we communicate about what we are going to do or what we are doing. Since we have a picture of the application in our mind, the communication process is viewed as a mixture of communication and the actual application. We can also say it is a mixture of the overall communication with the application.

The model above to the right is the same as the one to the left. In the one to the right, we show the input and the output connections of the communication process. The number of inputs to the left related to the number of people in the communication; see the entity usage section for more information about using the communication process entity. The table below provides more information about the inputs.

Inputs Left	Input Top	Input Bottom	Output Right
Communication	Application	Feedback	Communication and Application

The communication process entity connects the communication and the application. The communication process entity is used when we segment our application to show the actual application from our communication. The communication process connects the application and the parties involved in the communication. The output of the communication process connects to the input of the communication function. The communication process entity is similar to the communication application mixture entity. When using the communication process entity, all the parties that are involved in the communication can be considered as inputs. They can be connected directly; however it is always better to provide two inputs to that entity, the communication input and the application input; see the example section for more information about using this entity. The table below shows the output of the communication process entity, related to the communication input and feedback.

Input	Feedback	Output
Green	None	Green
Red	None	Red
Red	Blue	Green

Available Options

Available options for the communication process entity include

- Communication process
- Communication application mixture
- Etc.

The Analysis Window Entity

Analysis Window

Usage and Description

The analysis window entity can be viewed as two entities connected together. It can be viewed as a communication entity with feedback. The way to look at it, the feedback does not show, however it is already included. For example, we can say it is a communication entity, with the principles entity as feedback. The analysis window entity takes communication as input. The output can be connected directly to the application entity to provide an error free application. The table below shows the output of the analysis window entity related to the input.

Input	Output
Green	Green
Red	Green

Available Options

Available options for the analysis window entity include

- Error Correction Function
- ECF
- Error Correction
- Analysis Window
- Error Removal
- Etc.

The Sentence Analysis Entity

Sentence Analysis

Analysis

Usage and Description

The sentence analysis entity is the same as the principles entity. The sentence analysis entity is a feedback entity, it provides an error correction mechanism when input to another entity. For example, the sentence analysis entity can be connected to the communication entity, the word entity, etc. to provide feedback or error correction mechanisms. When connected to other entity, the sentence analysis entity provides error correction mechanism to enable them to provide an output without error. For example, if the entity is connected to the communication entity, it allows the output of that entity to be error free. The table below shows input output relationship for the entity connected to the sentence analysis entity.

Input	Feedback	Output
Green	None	Green
Red	Blue	Green

Available Options

Available options for the sentence analysis entity include the following

- Sentence Analysis
- Error Analysis
- Analysis
- Communication analysis
- Error Correction Function
- ECF
- Principle
- Etc.

The Communication Result Entity

Communication
Result

Usage and Description

The communication result entity is the result of our application. We can call it communication result we can also call it application result. If we segment our application into several steps to analyze its flow, the communication result is actually the result of the application. The communication result entity always follows the communication function entity. The

communication result entity is the output of the communication function entity. We can call it the result of the application or the result of the application execution function. Given that our application result depends on communication, the communication result entity also depends on communication. With that, the communication result entity is related to the communication input as provided by the table below. The output of the column of the table is viewed as the communication result, where the input is communication.

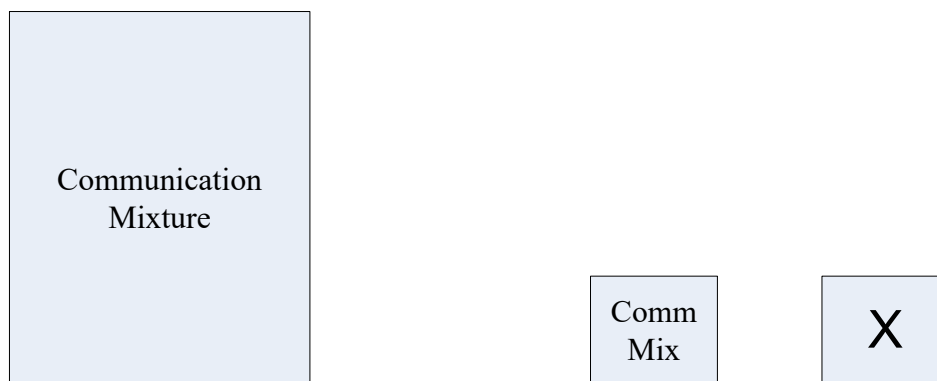
Input	Output
Green	Green
Red	Red

Available Options

Available options for the communication result entity include

- Communication result
- Application result
- Result of what we do
- Result of our communication
- Result of our application
- Project result
- Etc.

The Communication Mixture Entity



Usage and Description

The communication mixture entity is used to group people who are involved in the communication. Since the application involves communication, we can also say the

communication mixture entity is used to group people who are involved in the application. The person entity can be connected directly to the communication mixture entity, and then the communication mixture entity can be connected directly to the communication process entity. Depending on the application, the communication mixture entity can also be used to feed other communication mixture entity; see the entity usage section for more information about using the communication mixture entity. The table below shows the input output relationship of the communication mixture entity. The table below did not take feedback into consideration; however, with feedback, the result will be equivalent to the application process entity.

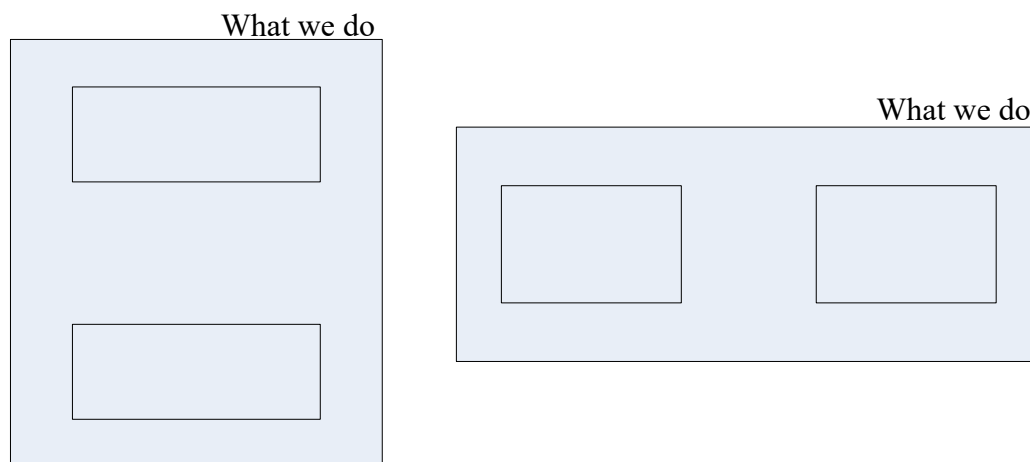
Input	Output
Green	Green
Red	Red

Available Options

Available options for the communication mixture entity include

- Communication Mixture
- Communication process—to some extent
- Comm Mix—shot name
- Etc.

What We Do Entity



Usage and Description

The what we do entity is simply another way to look at the application entity. The way to look at it, this entity is made of the communication entity and the application entity. It provides an effective way for us to separate our application from communication. This entity provides us the ability to separate what we do from communication. We can either use the top down version or the left to right version. In the top down version, the top entity is considered to be our communication, and the bottom one is our application, which is what we do. In the left to right version, the left entity is considered to be our communication, while the right box is our actual application or simply what we do. The table below shows the result of what we do related to communication.

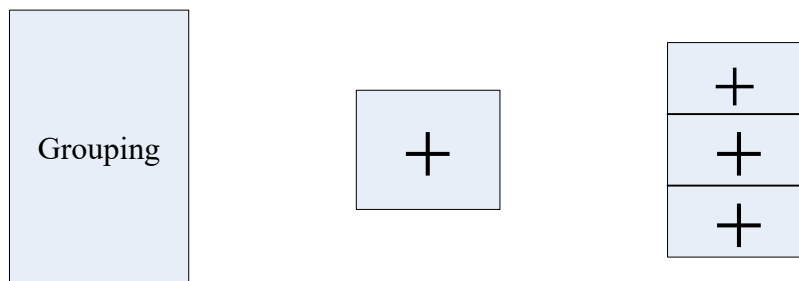
Communication	What we do
Green	Green
Red	Red

Available Options

Available options for this entity include

- Application
- Our application
- What we do
- What I do
- What you do
- Etc.

The Grouping Entity



Usage and Description

The grouping entity above is use to group entities like, for example, we can use the grouping entity to group people involves in the application. If our application is made of several parts, we can also use it to group parts of the application. We can use the plus sign or the one with the grouping word, it does not matter. The plus sign with multiple pluses can also be used to denote quantity involved in the grouping. Entities like or types can be attached to the grouping entity. Several parts that make up the application can be attached to the grouping entity. Also, mixture of communication can be attached to the grouping entity; see the example section for more information. When communication or communication mixture are attached to the grouping entity, the output of the grouping entity depends on that communication; see the table below for more information.

Input	Output
Green	Green
Red	Red

Available Options

Available options for the grouping entity include

- Grouping
- Addition
- Group
- Plus
- Etc.

The Error Entity



Usage and Description

The error entity can be used to separate errors in our communication. In other words, this entity represents the error part of our communication. To show the separation of error in our communication, the error entity can be used to identify that error. For example, if we identify a bad word in our communication, the error entity can be used to identify that word. When the error entity is identified in the communication, the result of the communication which is the application result is related by the table below. In this case, error is identified as red.

The error entity can be used to show errors in a typical communication. For instance, in a sentence, if a word is flagged as an error, the error entity can be used to identify that word. In a sentence that contains error, the error entity can be used to show the error. Given that the analysis process of a sentence involves both the content of the sentence and the purpose of that sentence, the error entity can be used to show both of them when there is an error.

Error	Result
Red	Red

Available Options

Available options for the error entity can also be anything that includes in the communication that causes the error. We mean the identity of anything in the communication that causes the error.

The Compensator Entity



Usage and Description

The compensator entity is considered to be an extra added to a communication in order to make the communication portable. For instance in a non-portable sentence, if one word is flagged as an error, if we remove that word and replaced by another word in order to make that sentence portable, the replaced word is viewed as a compensator. Since the analysis of that sentence involves both the function of that sentence and the content of that sentence, if we add an extra to that sentence to adjust the function of that sentence, that extra we add is considered to be a compensator as well.

Available Options

Available options for the compensator entity include anything added to a communication in order to make that communication portable. We mean the identity or the extra that we add to the communication to make it portable.

The Problem Entity



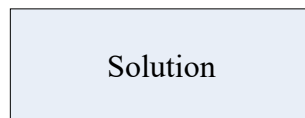
Usage and Description

While we use a circle with the give rise arrow to show a problem development, as an entity itself, we can use the entity above to show a problem. In this case a problem is an entity like an error is also an entity.

Available Options

Available options for the problem entity include the name of the problem or whatever other names we want to call it.

The Solution Entity



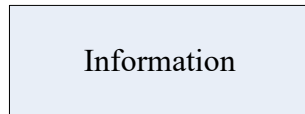
Usage and Description

The solution entity is used to identified a solution. The solution entity is used to show the solution of an identified problem. Usually, the solution entity is identified in the problem statement of the application. A lot of time, the solution entity must be used in order to solve the underlying problem. Refer to the example section for more information about using the solution entity.

Available Options

Available options for the problem solution include the name of the solution or whatever other names we want to call it.

The Information Entity



Usage and Description

The information entity can be used to show the content of the communication. This entity can be connected to other entities. Error analysis can also be performed on the information entity to determine the correctness of the information. In that case, the information entity can be connected to other entities. For example, the error analysis entity can be connected to the information entity, to determine the correctness of the information. In this case, the result of the information is related to that table.

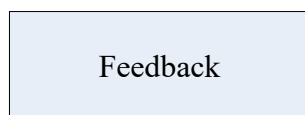
Feedback	Information
Blue	Green
None	Red

Available Options

Available options for the information entity include

- Information
- History
- Story
- Novel
- Article
- News
- Etc.

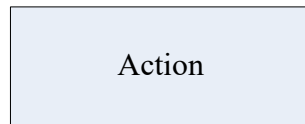
The Feedback Entity



Usage and Description

While the feedback entity may not be needed, however it can be used if necessary to show a given or specific feedback. The feedback entity is simply principle given that can be applied to correct a specific error. The feedback entity is the same as analysis, principles, and ECF.

The Action Entity



Usage and Description

The action entity can be used for problem statement. For instance, before we start our application or project, we can use the action entity flow connection to show our problem statement. The action entity can be used to show our problem statement. For example, to use the action entity, we can attach a person entity to its input, and the output can be attached to the reason entity. The action entity can also take analysis as feedback to determine the result of the action. We can feedback the action entity with the error analysis entity to make sure the action result is error free. The table below shows the output of the action entity, related to the input and feedback.

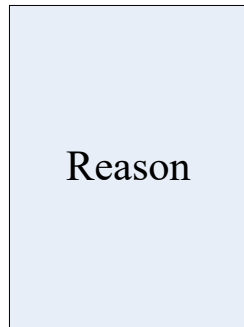
Input	Feedback	Output
Green	None	Green
Red	Blue	Green
Red	None	Red

Available Options

Available options for the action entity include the following

- Action
- Function
- This, That
- Etc.

The Reason Entity



Usage and Description

The reason entity is to be used with the action entity for a problem statement. For instance, the reason entity can be used with the action entity to determine the result of the action. The reason entity is considered to be the output of the action entity. We can also say that the reason entity is the result of the action entity. The reason entity answers the following question; why are we taking this action? Why are we performing this action? What problem do we solve by performing this action? The reason entity which is the output of the action entity is the result of the communication and the feedback in the action entity; the table below shows the result. Rather than using communication to show the result, in this case we use action and result; where the action determines the reason.

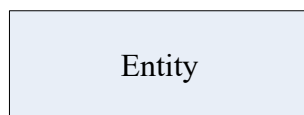
Action	Reason
Green	Green
Red	Red

Available Options

Available options for the reason entity include

- Reason
- Solve Problems
- Result
- Provide Solutions
- Etc.

The Entity Element



Usage and Description

The entity element can be used to represent an entity. It can also be used to represent elements. Another way to look at it, we can use the entity element to represent a physical entity. The entity element can be used to represent any actual entity. As we have just said, the entity element can be used to represent entities or elements. For instance we can use the element to represent a physical entity. We can also use it to represent a nonphysical entity. By representing an entity, we can simply put the name of that entity in the element.

Since the function of an entity is also an entity, it is possible to use the entity element to show an *entity as a function*. For instance, if we identify a function, then we can use the entity element to show that actual function. In this case, we can have something like this. Assume that we identify *Function 1*, and then *Function 1* is viewed as a function of an entity. In this case, we can show it like the model below, where *entity 1* is viewed as the actual entity and *function 1* is viewed as the actual function.



Since the aspect of an entity is also an entity, the entity element can also be used to show the aspect of an entity. In this case, it is being used to show an *entity as aspect*. Assume that we identify an entity, since the aspect of that entity is also an entity, then we can use the entity element to show that aspect as follow: Let's say we identify *Entity 1*, where *Entity 1* has *Aspect 1*, and then we can use the entity element to show that in the form below. In this case, *Entity 1* is the actual entity, where *Aspect 1* is the actual aspect of that entity and it is also an entity.



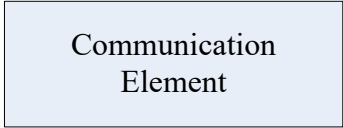
Available Options

Available options for the entity element include

- Entity
- Element
- Physical entity
- Nonphysical entity
- Name of the entity

- Item
- Thing
- Any thing
- Function
- Function of entity
- Aspect
- Aspect of entity
- Etc.

The Communication Element Entity



Communication
Element

Usage and Description

The communication element entity represents a communication entity. We can say that the communication element represents communication entities that are not a word, a sentence, and a paragraph. While we use the term communication element, we can also use the term communication entity as well. The communication element entity can be used to represent communication entity. For instance, we can use the communication entity element to represent pictures, images, videos, drawings, diagrams, models etc.

While we say that the communication element entity is used to represent communication elements that are not words, sentences, and paragraphs, however since words, sentences and paragraphs are considered to be communication elements, the communication element entity can be used to represent them as well.

Available Options

Available options for the communication element include

- Picture, photo
- Image
- Video
- Drawing, diagram, model

- Chart, flow chart
- Graph, table
- Audio
- Voice
- Word, sentence, paragraph
- Book, journal, newspaper, magazine, video, video,
- Etc.

The ECF Entity



Usage and Description

This short ECF entity version is considered as an error correction entity. The way to look at it, this entity can be connected to other entities to send a correction signal to that entity to enable an error free output. As we say above, the ECF entity can be connected as feedback to other entities to send signals to them to enable them to correct any error that is presented. For example, this entity can be connected to the communication process entity to enable the output of the communication process entity to be error free. When this entity is connected to an entity, it sends a blue signal to that entity. The table below shows the input output relationship of the connected entity related to the ECF entity. From the table below, ECF None means no presence of ECF.

Connected Entity	ECF	Output
Green	None	Green
Red	Blue	Green
Red	None	Red

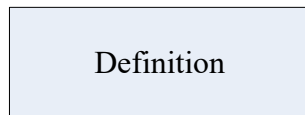
Available Options

Available options for the ECF entity includes

- Error Correction Function
- ECF
- Analysis Logic
- Principles

- Sentence Analysis
- Error Correction
- Parent Principles
- Etc.

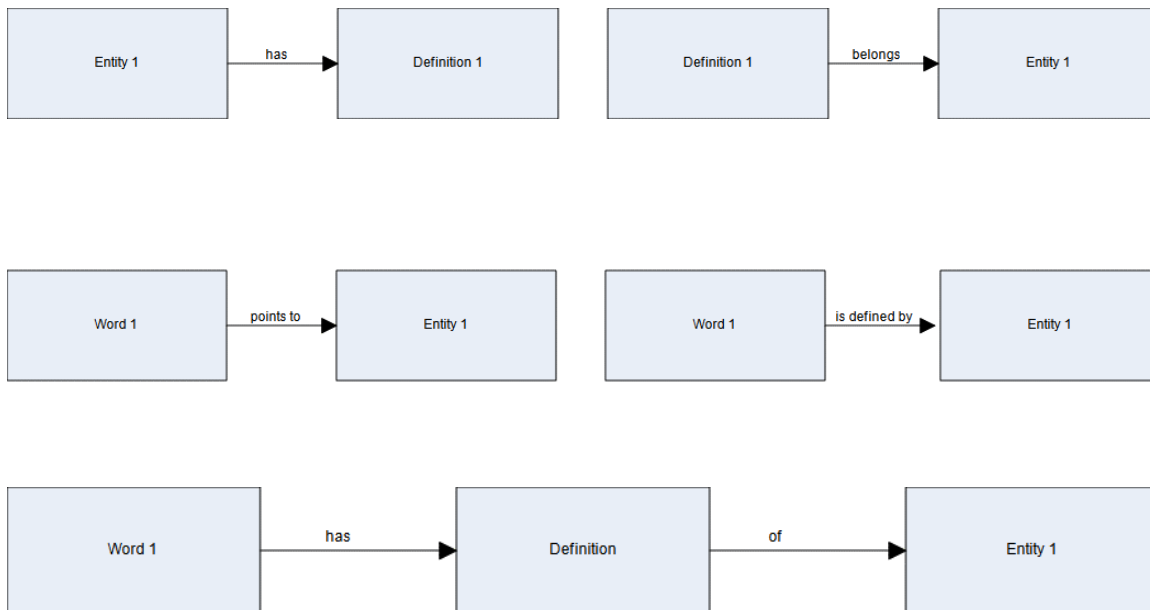
The Definition Entity



Usage and Description

By understanding the entity element, the word entity, and the information entity; the definition entity may not be needed and it is not needed in our model. Since word 1 points to entity 1 and definition 1 points to entity 1, so it is not needed to show the definition entity in a model.

We can use the definition entity to identify the definition of an entity. For example, the definition of an entity belongs to that entity; we simply identify that definition. Below are some examples of using the definition entity.



The Dictionary Entity

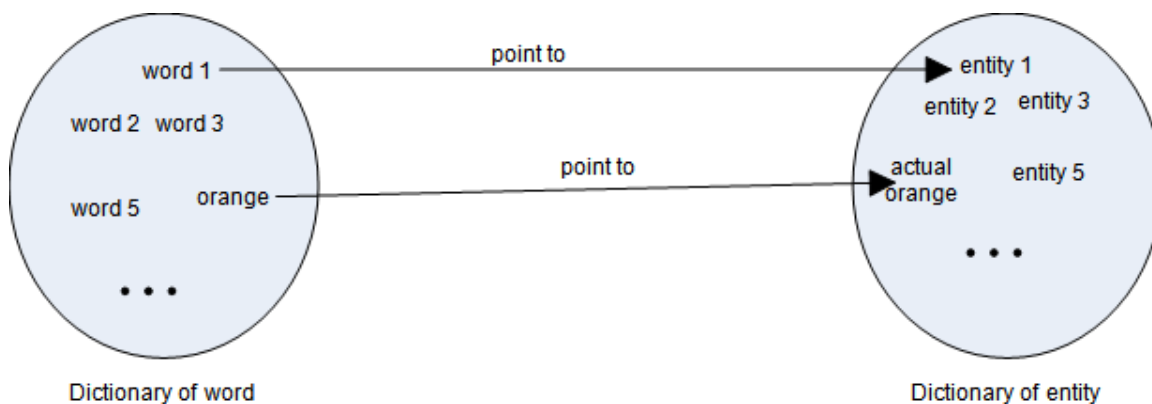


Usage and Description

The dictionary entity can be used to identify a dictionary. The dictionary entity can be viewed as a collection of words. It can also be viewed as a collection of entity.



We can map a word to an entity to identify the entity the word points to. Here is an example that maps words from the dictionary to their respective entity from the entity dictionary.



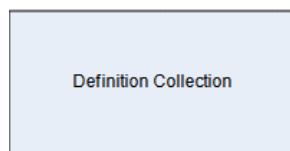
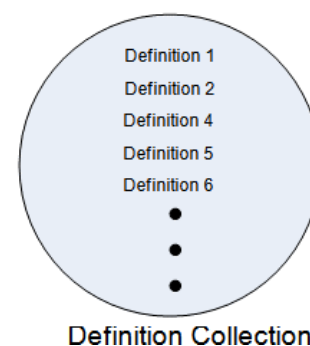
The dictionary of word and entity do not represent books, they represent actual entities. We can think of the dictionary of words as a collection of existing words like orange, book, chair, table and so forth and the dictionary of entity as actual existing entities like actual book (for example, the one next to me), actual table (for example, the one I am using now), actual chair (for example, the one I am sitting now) and so forth.

Available Options

Available options for the Dictionary entity includes

- Dictionary
- Dictionary of word
- Dictionary of entity
- List of words
- List of entities
- Collection of words
- Collection of entities
- Group of words
- Group of entities
- Set of words
- Set of entities
- Words
- Entities
- Etc.

The Definition Collection Entity



Usage and Description

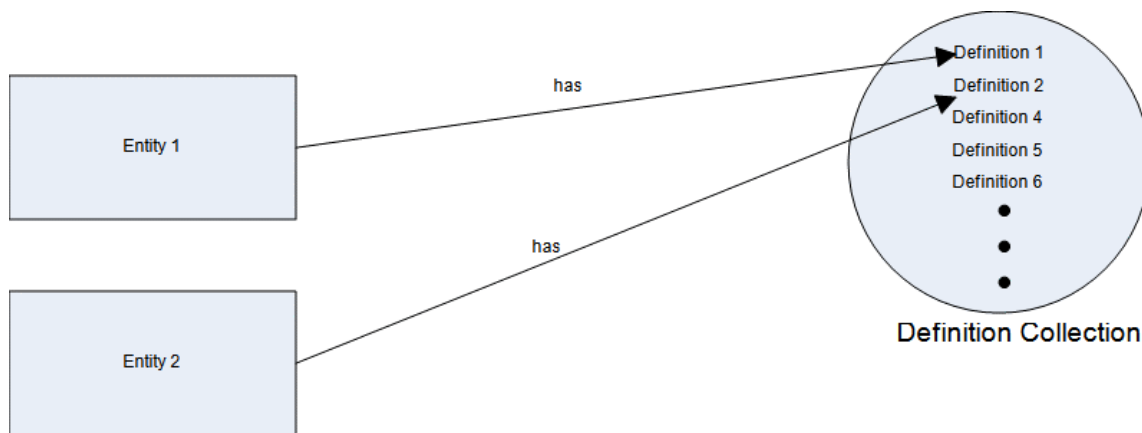
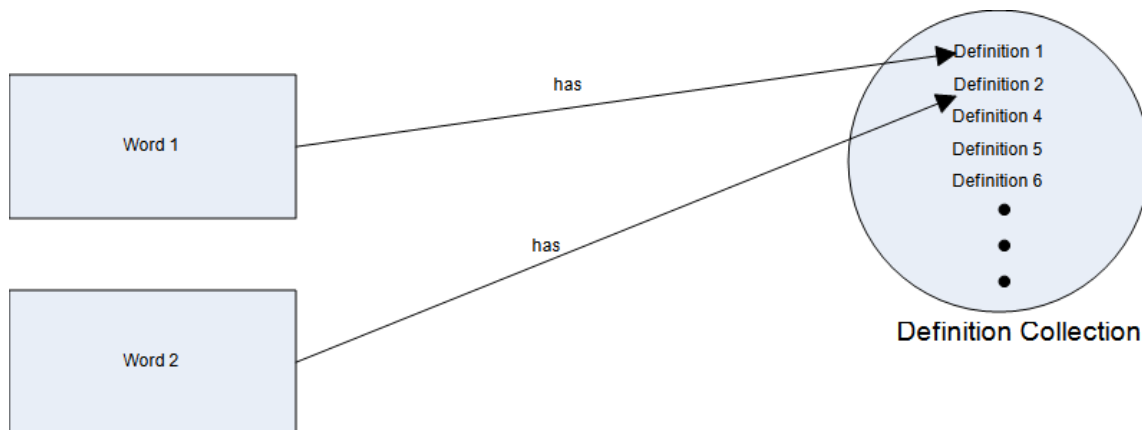
We use the definition collection to identify a collection of definitions. We can think it as a dictionary of definitions. We can look at it like this. The definition of an entity is a separate

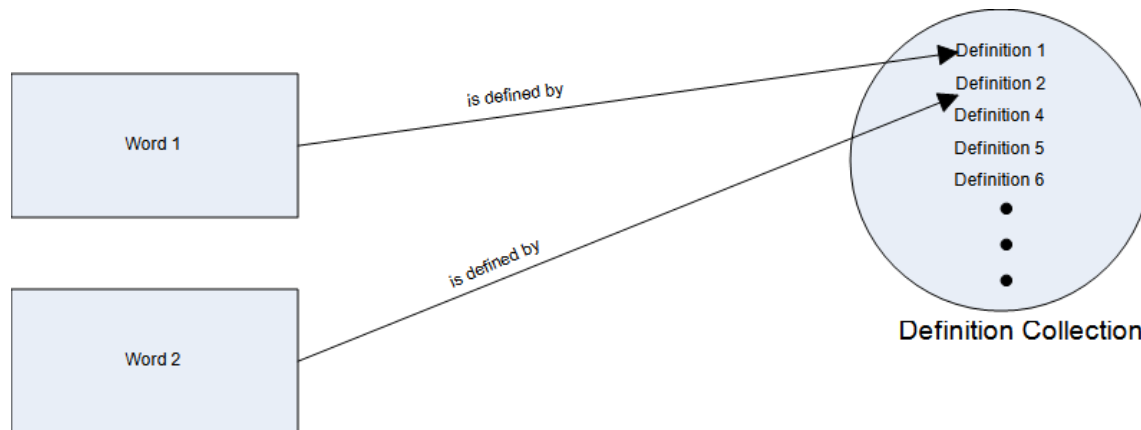
entity from that entity, where the definition collection entity contains a collection of definitions. We can think it as a list of definitions.

Available Options

Available options for the Definition Collection entity includes

- Definition collection
- List of definitions
- Dictionary of definitions
- Set of definitions
- Definitions
- Etc.





The Separation Line Entity



Usage and Description

We can use the separation line to separate the steps of our application process. We can also use it to show the separation of our application from communication. The way to look at it, from start to finish, we can use this line to show the steps we undertake from the starting of our application to the finish of our application. We can use this line to show the separation and provide more information about each step.

We can also use the separation line to show entity separation, for instance word separation in sentences, sentence separation in paragraphs etc. After we have identified some entities within a group, we can use the separation line to separate each entity. After we have identified entities that make up our communication, we can use the separation line to separate the entities or communication elements. Using the separation line is important, since it enables us to concentrate in the area of our interest during the analysis.

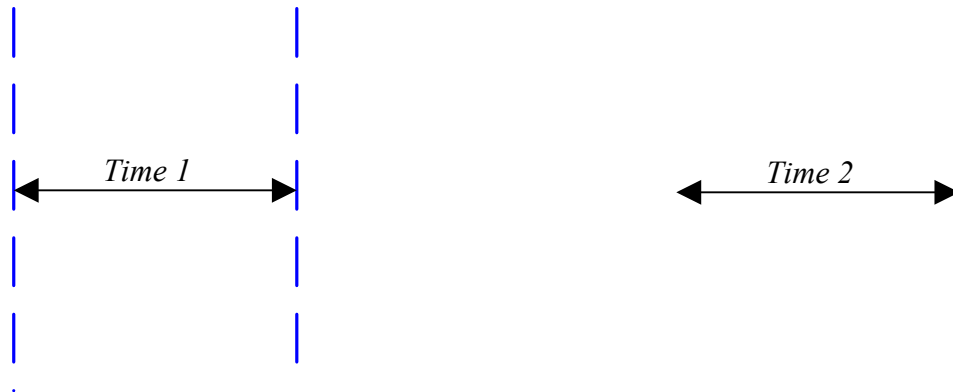
The separation line can also be used to show steps in a relationship model. For instance, we can use the separation line and the time line to show steps in a relationship model related to time. Refer to the example section for more information.

Available Options

Available options for the separation line include

- Separation
- Process separation
- Step separation
- Domain separation
- Communication and application separation
- Entity separation line
- Element separation line
- Item separation line
- Etc.

The Time Line Entity



Usage and Description

Similar to the separation line, if desired time line can be used to show function execution or specific event related to time. For instance, we can use timeline to show specific communication at a given time; see the example section for more information. Above, the model to the left shows the first timeline, while the one to the right shows subsequent timeline.

Available Options

Available options for the time line include

- Time line name
- Date
- Time
- Subsequent timeline or subsequent line
- Date line
- Etc.

The Progress Entity



Usage and Description

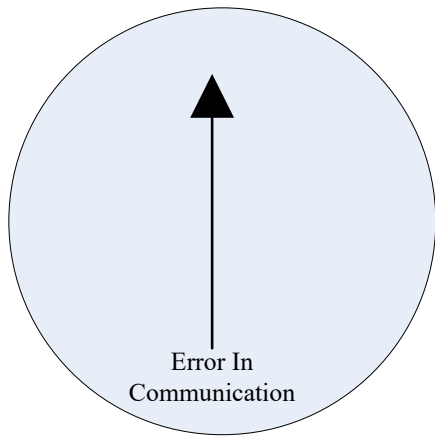
The progress entity or progress line can also be used to show the progress of a function or the progress of our communication in our project. We can use the progress entity to show the progress of our application function execution. We can also use the progress bar to show the progress of the overall project we are working on.

Available Options

Available options for the progress bar include

- % of completed on time
- Progress related to time
- Progress line
- Progress entity
- Etc.

The Error in Communication Gives Rise Entity

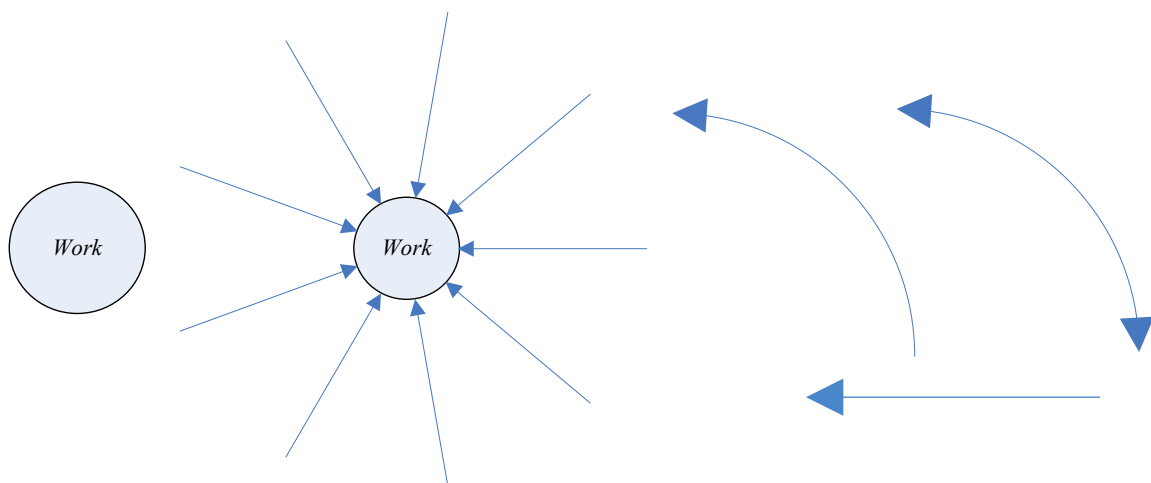


Description and Usage

Given that our application is communication driven and when we commit errors in communication it affects our application, this entity or model can be used to show the result of our application related to errors in communication. This model can be filled to show the result of the application based on errors. By tracking errors in our communication, if there is a problem in our application, at the end we can use this entity to show that problem.

The Work Entity

This entity can be used to construct the communication related to work model. Since our project is communication driven, we can show how people work together related to communication. In this case, it makes sense for us to show that in a circular form. By using the work entity with the arrows, we can construct a circular model to show how we work together from our communication.

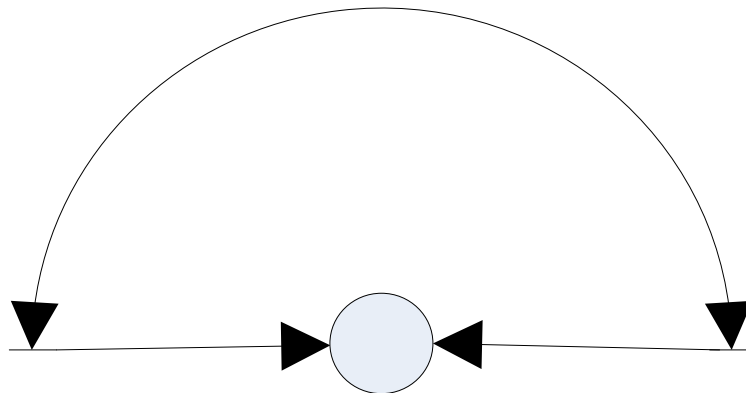


Available Options

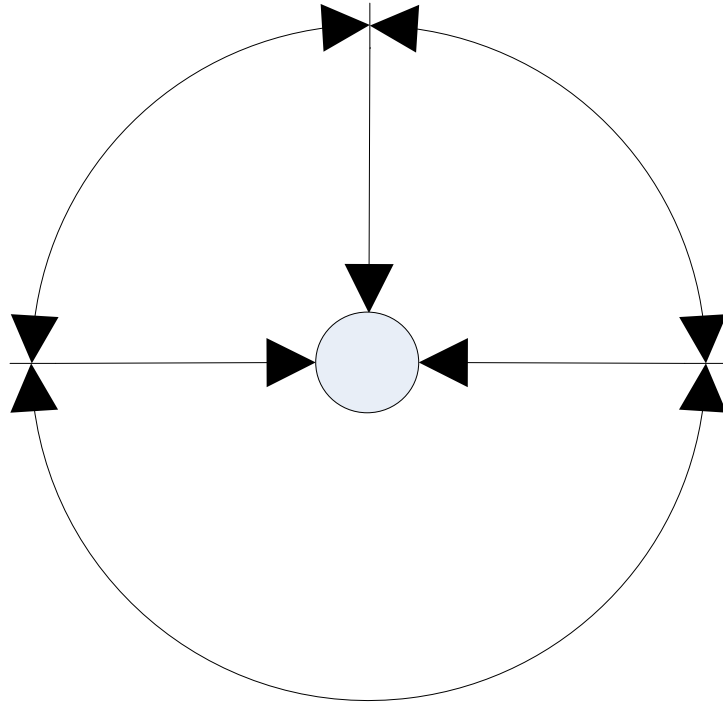
Available options for the work entity include

- Work
- What we do
- Etc.

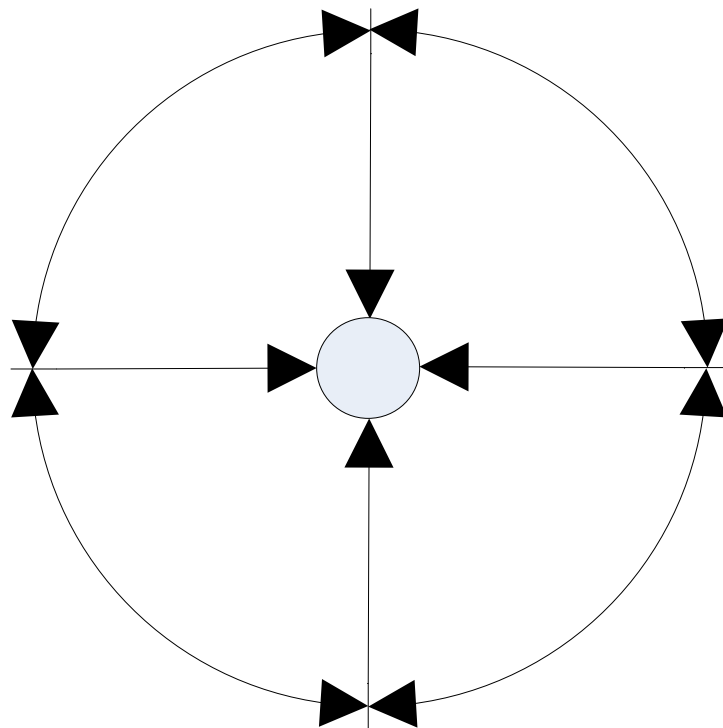
Assume that we have two people working in a project, we know that those two people communicate together to do their work. In this case, we can use the work entity to construct a model to show the flow of the communication of those people related to what they do. Using the model below, we simply show two people communicating together to do what they do.



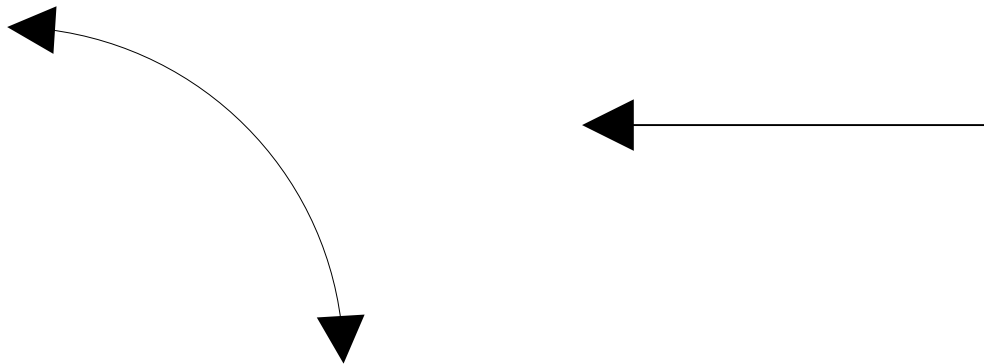
Using the same model above, we can add another person to the project. In this case, we have three people working together. The communication of these people is what enables them to work together. In this case, we can simply show that in a circular form as shown below. The model below shows communication of three people related to work.



Assume that we have four people communicate together to do their works, we can then extend the work together model to show that. In this case, we can show the people communicate together to do their works as shown below.



We can simply name the two arrows below communication related to work arrow. We use the arc arrow to show communication from person to person, while we use the straight arrow to show what a person does in relation to communication. The communication related to work arrow can be used to construct the model of the communication related to work and to show the flow of the communication in that model. As we previously said, the arrow to the left shows the flow of communication from person to person, while the one to the right shows the flow of the communication related to the function or the purpose of the communication. In other words, the one to the right shows the flow of the communication related to what we do.

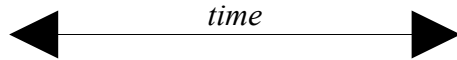


Time Chart and Dateline Entity

Time line can be used in conjunction with dateline to record events, statements, questions etc. Below we show a timeline and a dateline that can be connected together to show record of statement; see the entity usage section for more information.



The Time Entity



Usage and Description

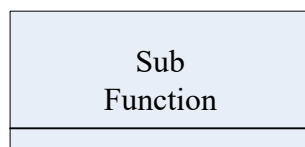
Assume that we are using the time chart with the date line where we want to show the difference between the two dates, we can use the time entity to show that. For instance, if we want to show how long it takes from the first date to the second date, we can use the time entity to show that. We use the time entity to show time or exact time between two dates or times.

The Operating Principle Line Entity

Usage and Description

The operating principle line entity can be used with the principle entity and the person entity to show the principle the people in an application depend on. Refer to the example section for more information about the usage of the operating principle line entity.

The Sub Function Entity

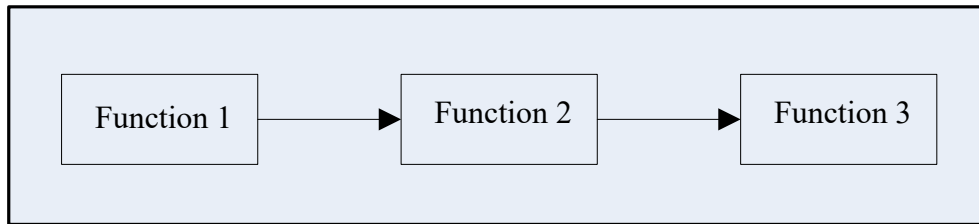


Usage and Description

It is always possible for a project to be divided into different sections, where the main function of the project is made of many other functions. In this case, it is always possible to have sub functions. A sub function can be considered as a function that is a part of the main function. It can also be considered as an external function. Assume that a project is divided into several sections, where each section is managed by a different person; it is possible for each of those sections to be set as a sub function. While smaller groups of people tend to be easier to manage,

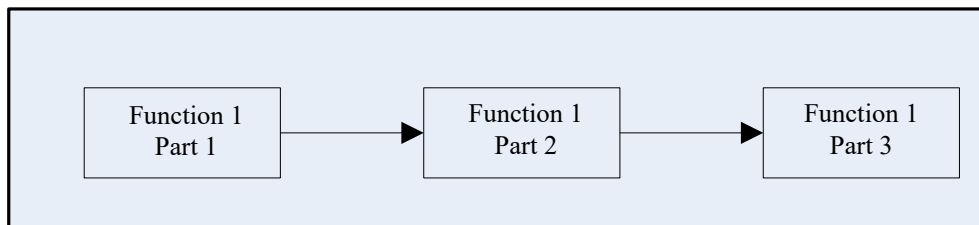
using sub functions is considered to be a better way of managing a project. The way to look at it, if the communication includes several functions where some of those functions are embedded inside other functions, the sub function entity can be used to show functions that are included inside other functions. Below shows the communication function and it is made of tree functions.

Communication Function

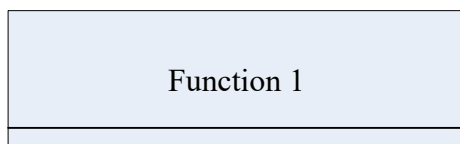


The model below shows *function 1* which is a part of the communication function and it contains other functions inside.

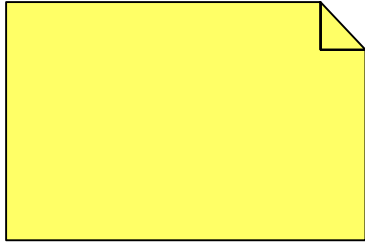
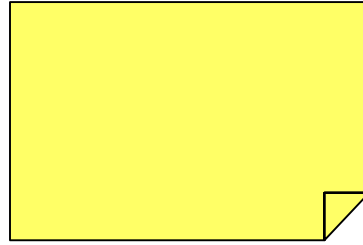
Function 1



Functions that are considered as sub functions can be identified as the following; in this case, *function 1* contains other functions inside. We use the extra line inside *function 1* near the bottom to indicate that *function 1* is a sub function or *function 1* contains other functions inside.



The Notes Entity



Usage and Description

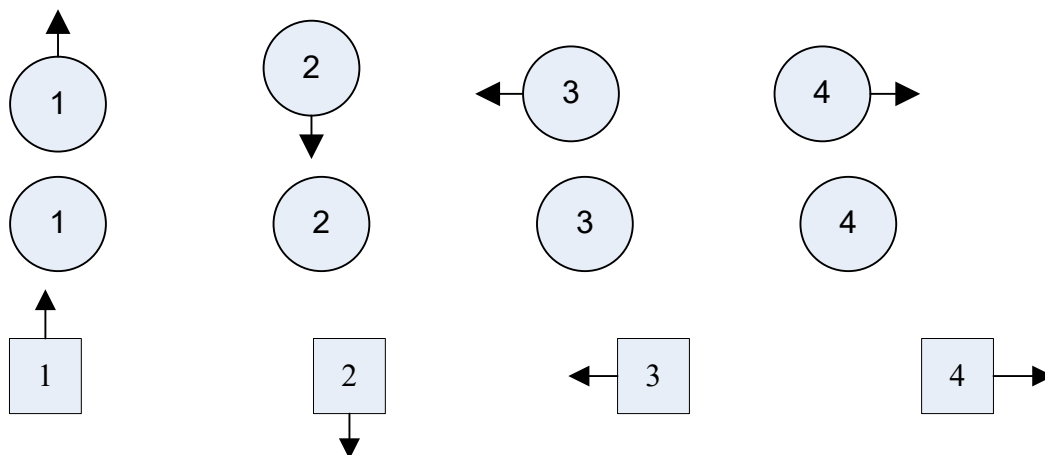
When using a computer, sometime it is necessary to take notes. For instance, within a project screen on a computer, it may be necessary for a person to use a sticky note as a reminder; in this case, the note entity can be used for that. The note entity can be used on a computer screen to show user's notes. Both of the entities shown above are the same. Users can use them to show or hold their notes. The note entity can also be used on paper or board if desired. For instance, within our project, we can use it as a reminder.

Available Options

Available options for the note entity include

- Note
- Note entity
- Note holder
- Sticky note
- Etc.

The Node Entity





Usage and Description

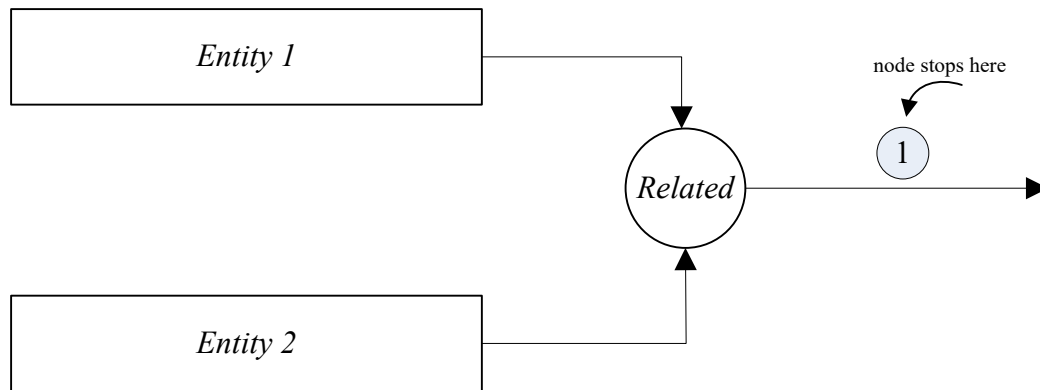
Within a communication or conversation, a node is considered to be an important point of that communication or that conversation. We can also say that a node is a point of interest of a communication. We can flag specific point in a communication and assign a node to it. From the model above, we show the nodes both in square form and circle form. As shown above, nodes can also be shown with or without arrow as longer they are placed properly. Refer to the entity usage section for more information about using the node entity.

Available Options

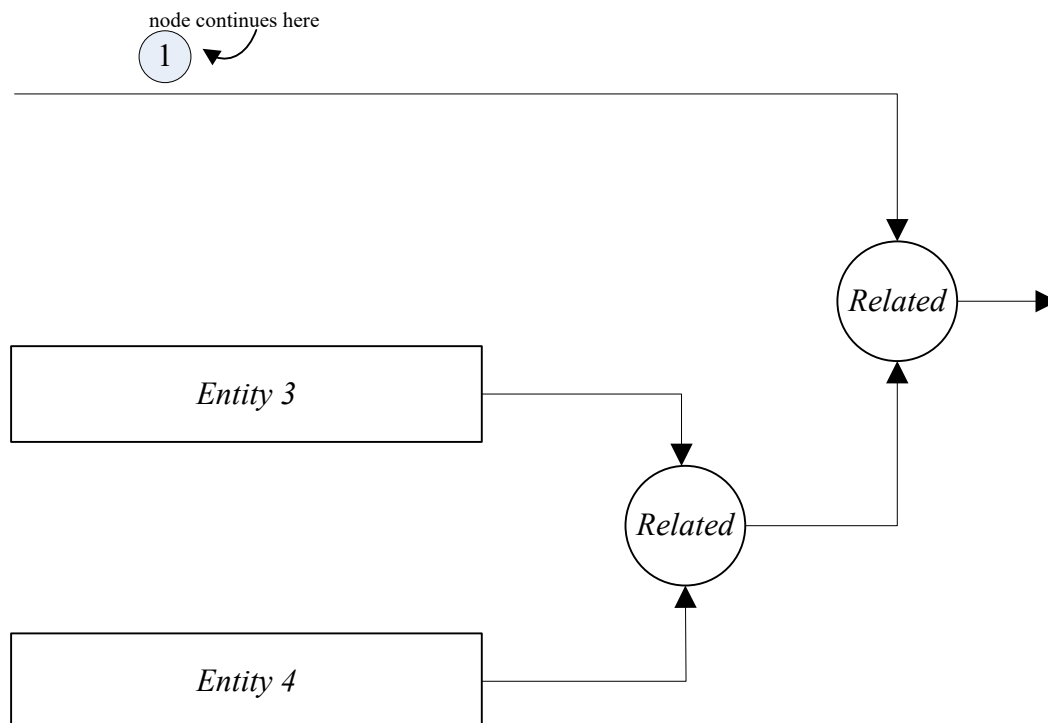
Available options for the node entity include

- Node
- Node number
- Point
- Point number
- Location
- Location number
- Communication node
- Information node
- Specific location
- Etc.

Nodes can also be used to show continuity from one point to another point. In other words, we can use the node entity to show continuity from one location to another location. For instance, we can use node number to show the continuity of a relationship or specific entity at a specific point. Let's assume that *Entity 1* is related to *Entity 2*, and that relationship is related to the relationship of *Entity 3* and *Entity 4*. Now, if in a piece of paper or drawing board we cannot fit all of it, then we can place a node number in that part, then continue by putting another node number to another piece of paper or in another part of the same sheet or in another part of a drawing board or another drawing board to show the continuity of that relationship as shown by this model.



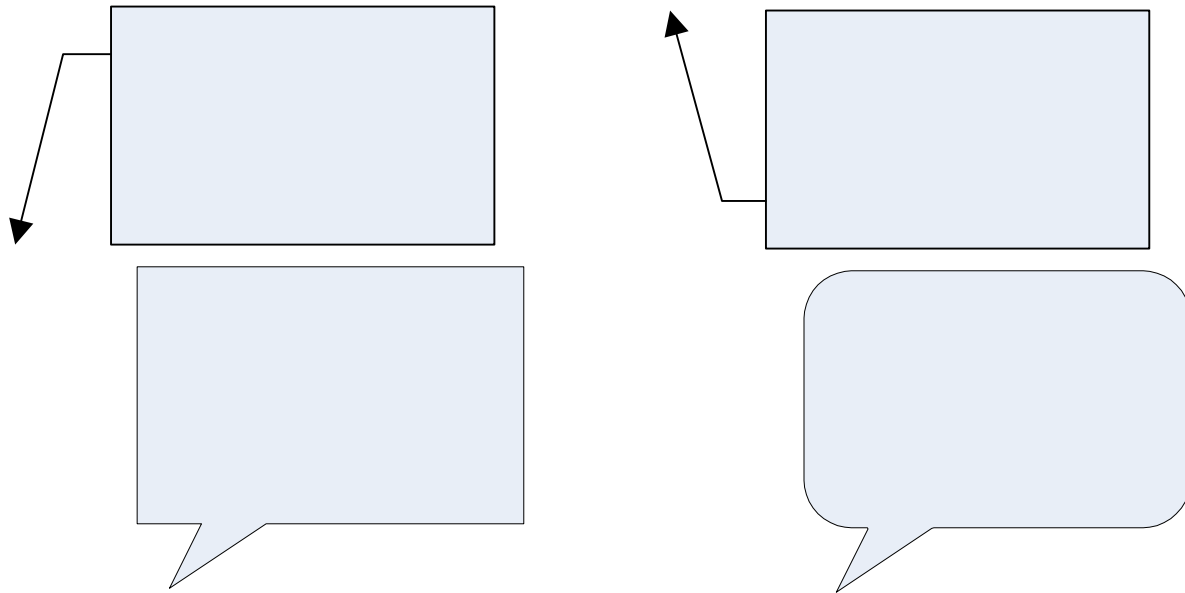
sheet 1 or part 1 of sheet 1



sheet 2 or part 2 of sheet 1

The node entity can also be used in a relationship to identify an entity at a specific node, where that entity can be defined and identified by that node. Refer to the example section for more information about defining and identify an entity at a node in a given relationship.

The Callout Entity



Usage Description

Callout can be used to show information at specific point. Instead of using the node entity to show node number then give information of that node, the callout entity can be used instead to show specific information at that point. Overall we use the callout entity to record sections of interest in our communication or conversation. Refer to the example section for more information about using the callout entity.

Available Options

Available options for the callout entity include

- Callout
- Callout entity
- Information point
- Node information
- Etc.

The Node Information Table Entity

Node Number	1	2	3
Node Information			

Usage and Description

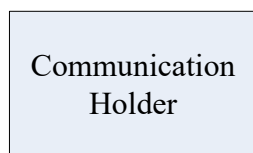
When using the node entity, it is possible for us to provide information about the nodes we identify. By using the node table, we can provide information about the nodes. The node table information is made of node numbers and node information. For instance if within our project model we record node number 1 in our communication, it makes sense to provide more information about node number 1 or the communication at that point. By using the node table information, we can provide more information for that node. Refer to the example section to learn more about using the node information table.

Available Options

Available options for the node information table include

- Node information table
- Table
- Information table
- Information table at specific locations
- Etc.

The Communication Holder Entity



Usage and Description

As we have learned previously, our project can be divided into different sections, where each section is managed by a different person. When that happens, it may be possible for an employee in a project to work in two sections. For instance in a project, *function 1* is considered to be the main function for *employee 1* and that employee is managed by *manger 1*. Within that same project, *manager 2* is managing *function 2* and needs some help from *employee 1*. Now, *employee 1* is working in both *function 1* and *function 2*, but managed by both *manger 1* and

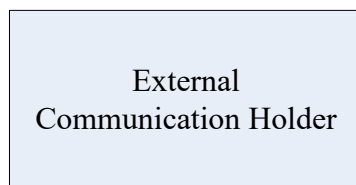
manger 2. We assume that *function 2* is a sub function within the project. When viewing on the screen or paper, it is not practical to show double representation of *employee 1*. In this case, if the sub function is open within the main function, then *employee 1* is replaced by his/her respective communication holder in *function 2*. It is very important to understand the communication holder entity and look at it in a practical approach. In life a person cannot be duplicated. It is not possible to see double representation of a person. For this reason, the communication holder is important, since it allows one person to work on multiple sections of a project, but can never be seen or viewed as duplicate. Refer to the example section to learn more about using the communication holder entity.

Available Options

Available options for the communication holder entity include the following

- Communication Holder
- Information Holder
- Conversation Holder
- Etc.

The External Communication Holder Entity



Usage and Description

Since the result of our project is a function of communication, it is very important for us not to take communication for granted. While working in a project, it may be possible for us to use communication or information from people that are not physically in that project. By using the external communication holder, we make it possible to use communication from people that are not physically in our project. This is the way to look at it. Assume that we are working in a project where we need information to accomplish what we are doing. The information we need can be from a person who is not in the project, a magazine article, newsletter, etc. By using the external communication holder, we can include those entities in the project, but externally. In other words, we use those communications in the project, but they are not from people who are physically in the project. In this case, we simply represent them as external communication

holders. Refer to the example section to learn more about using the external communication holder entity.

Available Options

Available options for the external communication holder include the following

- Person name, people
- Article, magazine article, radio program, TV program, journal etc.
- Book, magazine, program, newspaper, newsletter
- Website, web page
- Email, mail
- Etc.

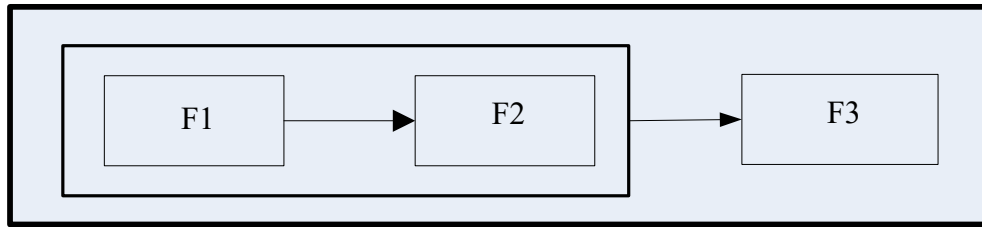
The Empty Container Entity



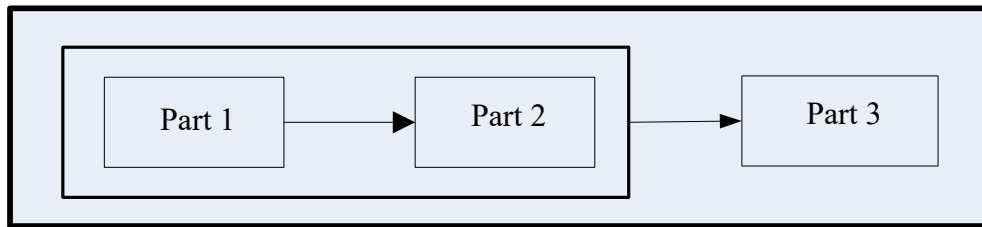
Usage and Description

The empty container entity can be used to group parts of the application, parts of the communication function, and parts of the application result. The first model below shows the usage of the empty container entity to group some parts of the communication functions. The second model shows the usage of the empty container to group some parts of the application, while the third model shows the usage the empty container to group some parts of the application result. In this case we assume the overall application result is made of different parts. Depending where we place the empty container, we can give it the following names: part of application container, part of communication function container, and part of application result container. Refer to the entity usage section form more information and example about using the empty container entity.

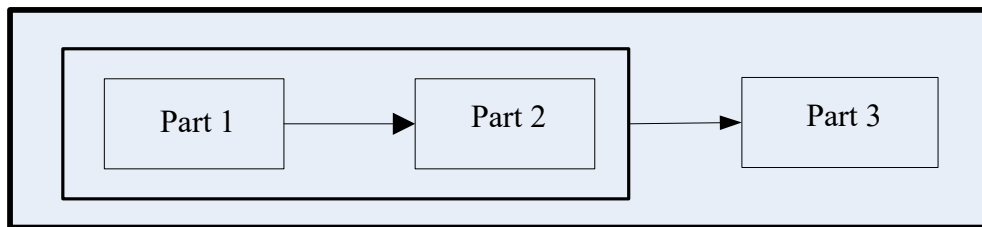
Communication Function



Application



Application Result

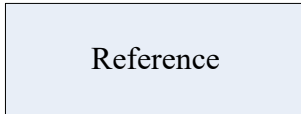


Available Options

Available options for the empty container entity include

- Empty container
- Part container
- Function container
- Result container
- Application container
- Empty container
- Etc.

The Reference Entity



Usage and Description

Within our project model, if it is needed, the reference entity can be used to show references. For instance in our project model, it might be possible for us to link to another page, in this case we can use the reference entity for that. The reference entity can also be used to provide reference information. Since not all entities are considered to be references, it may be possible for a reference to undergo the same analysis as a regular entity. Although references are principles themselves; but in order for that to happen, the principles must be understood. In the event that the principles are not understood, it is possible for an indicated entity that claims to be a reference, but not identified as a reference at all. To determine that, analysis can be used to analyze an entity to determine if it is a reference or not; as shown by the table below; refer to the example section for more information about using the reference entity.

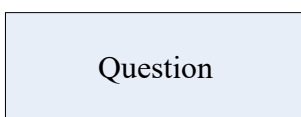
Input	Feedback	Output
Green	None	Green
Red	None	Red
Red	Blue	Green

Available Options

Available options for the reference entity include

- Reference
- Reference entity
- Reference information
- Reference information
- Link information
- Etc.

The Question Entity



Usage and Description

The question entity is used to show a question. Since questions are parts of communication, during our communication it is possible for us to ask questions to each other. The question entity is used during a communication to indicate a question. Since questions are parts of communication and they undergo the same analysis as communication, feedbacks can be given for an asked question. The following table shows the result of a question with and without feedback.

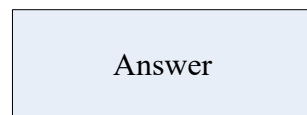
Input	Feedback	Output
Green	None	Green
Red	None	Red
Red	Blue	Green

Available Options

Available options for the question entity include

- Question
- Inquire
- Query
- Part of communication
- Communication element
- Communication entity
- Etc.

The Answer Entity



Usage and Description

The answer entity is used to show the answer of a question. Since answers of questions are part of communication, during our communication it is possible for us to ask questions and answer questions to each other. The answer entity can be used during our communication to indicate the answer to a question. Since answers are parts of communication, they undergo the same analysis as regular communication. During the answering of a question, feedback can be given to

determine the correctness of that answer. In this case, the feedback enable us to determine if the underlying entity is an answer or not. The following table show the result of an answer related to feedback.

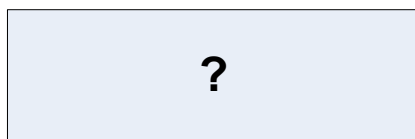
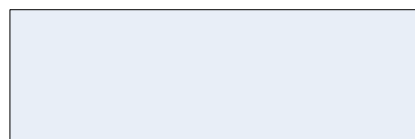
Input	Feedback	Output
Green	None	Green
Red	None	Red
Red	Blue	Green

Available Options

Available option for the answer entity include

- Answer
- Response
- Reply
- Part of communication
- Communication element
- Communication entity
- Etc.

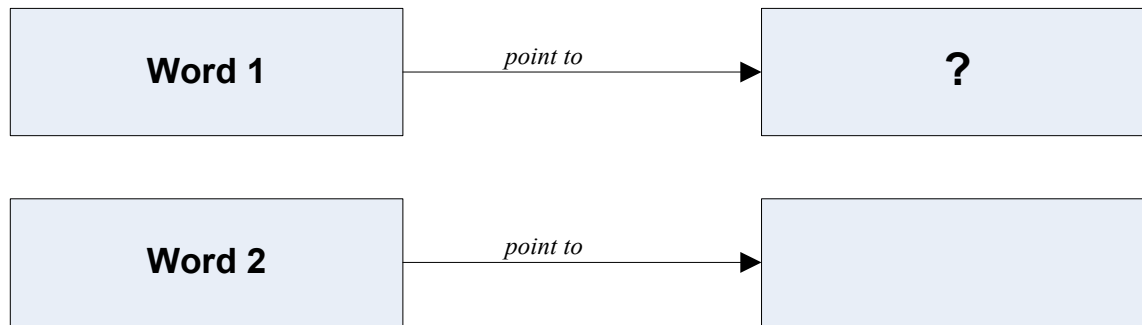
Unknown Entity



Usage and Description

Since we communicate relatively with entities that we identify, during our communications, it may not be possible for us to identify unknown entities. Since not all of us understand the principles of communication during our communications, it may be possible for some of us to talk or communicate about entities that we don't know—do not identify—or do not exist. When

that happens, it makes sense for us to analyze that communication, to ask questions about those entities or whether they point to actual entities. In this case, it is possible for us to request information on whether words that refer to those entities point to actual entities. In this case, we can have something like this.



As shown above, the first model means what entity *Word 1* points to, while the second model means *Word 2* points to a blank entity. Since there is no such as blank entity, in this case the blank entity refers to an entity that does not exist in the communication.

In addition to the way we have it above, it is possible for us to use cross an entity to show an entity that does not exist. By doing so, we can have the form below



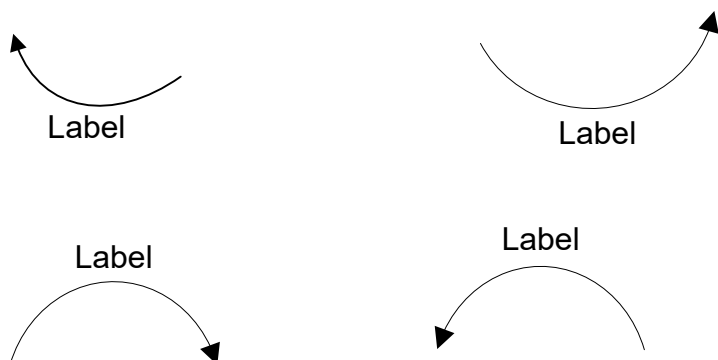
Number Identification

#1 #2 #3 #etc.

Usage and Description

During communication, it is possible for us to separate or identify an entity and refer to it by a number. In order to do that, we can use the # identification with any number to label an entity or refer to it.

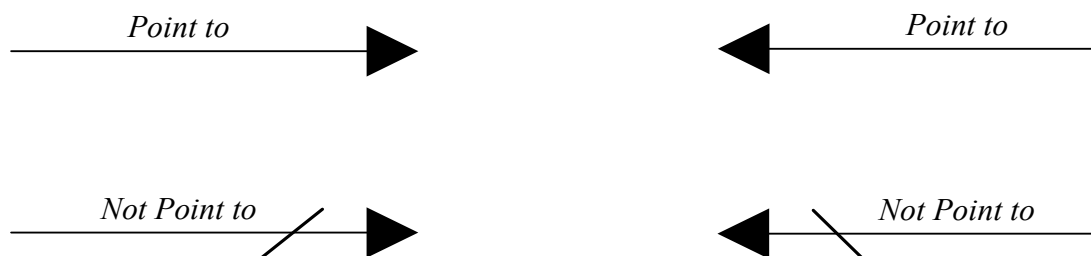
The Information Label



Usage and Description

The information label can be used to show more information about entities. The following labels can be used to show more information about entities identification. We can use them to add more information on entities or simply to identify more information of entities. The labels can be positioned to locations of our interest. For instance, they can be flipped and rotated to reflect a desired position.

The Point to Label



Usage and Description

Since when we see or repeat a word we think about what that word represents, the point to label enables us to identify an entity that is represented by a word. For instance, we use the point to label to show a word that is represented by an entity or physical entity. We can also use the point to label to show a communication entity that points to an actual entity. Refer to the example section for more information about using the point to label.

Available Options

Available options for the point to label include:

- Point to
- Define
- Define from
- Define by
- Map to
- Mean
- Identify
- Identify by
- Point to arrow
- Word point to entity label
- Word point to entity information
- Word to entity information label
- Point to entity
- Point to actual entity
- Show
- Etc.

The Give Rise Label



Usage and Description

The give rise label can be used to show an entity that gives rise to another entity. For instance, we have used the give rise arrow to show how a problem developed from an error in communication. In this case, we use it to show a problem development process where a problem is derived from a communication error.

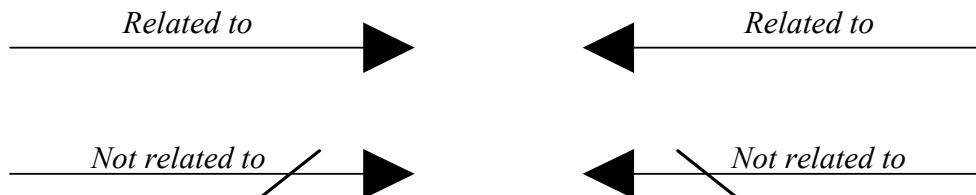
Available Options

Available options for the give rise arrow include:

- Give rise
- Derive from
- Develop from
- Derive by
- Give rise entity
- Become
- Cause
- Come form
- Turn to
- Etc.

The Relationship Label

We can use the relationship label to show an entity that is related to another entity. We can rotate the label to change its position accordingly. We can also change the text to reflect to what we are doing.



Available Options

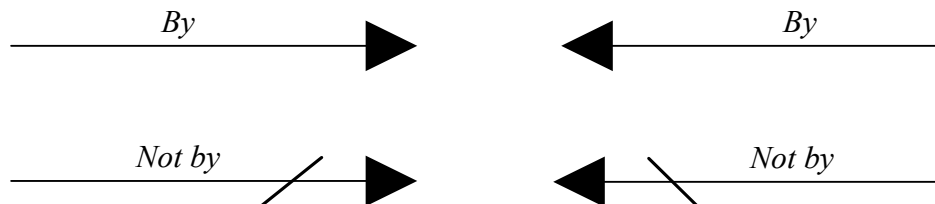
Available options for the relationship label include:

- Relate
- Link
- Connect
- Part of
- Has
- Contain
- Etc.

The By Label

We know that the relationship between two entities is also an entity; we can use the by label to identify that entity. In other words, we use the by label to identify the relationship between two

entities. Let's say it again; the purpose of this label is to identify entities that serve as relationships between other entities. For instance if entity 1 is related to entity 2 and we identify the relationship entity as entity 3, we can use the by label to identify that entity. In this case, we can simply put it in a form like entity 1 *related to* entity 2 *by* entity 3. The label can be rotated and flipped accordingly. The text can also be changed accordingly as well.



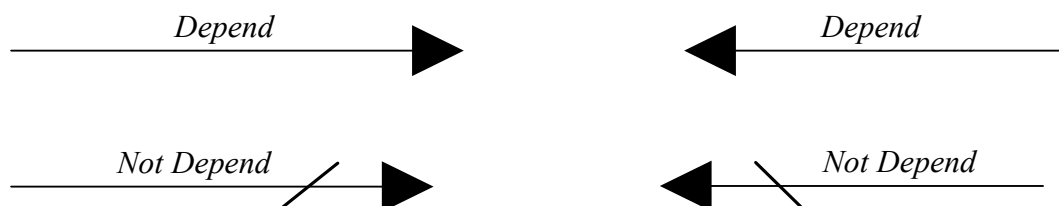
Available Options

Available options for the by label include:

- By
- By this
- By that
- Connected by to some extent
- Linked by to some extent
- Etc.

The Dependency Label

The dependency label is used to show an entity that depends on another entity. Since a given entity can depend to another entity, it is possible for us to use the dependency entity to show that. For instance, if *entity one* depends on *entity two*, then we can use the dependency label to show that *entity one* depends on *entity two*.



Available Options

Available option for the dependency label include

- Depend

- Related
- Dependency entity
- Etc.

The Agreement Label

The agreement label is used to show an entity that agrees with another entity. Since we communicate relatively about entities that we identify and the communications about those entities depend on them rather than us personally, it is possible for us to show the agreement of an entity with another entity during our communication. We can use the agreement label to show that. For instance, if a communication agrees with an entity, then the agreement label can be used to show that. If the answer of a question agrees with specific information, then the agreement label can be used to show that.



Available Options

Available option for the agreement label include

- Agree
- Match
- Related
- Etc.

The Match Label

The match label can be used to show an entity that matches with another entity. Since communication about specific information must match that information, during our communication about specific information, it is possible for us to use the match label to show that our communication matches the actual information.





Available Options

Available options for the match label include

- Match
- Related
- Agree
- Point to
- Go with
- Map
- Etc.

The Inclusion Label

The inclusion label can be used to show an entity that includes in another entity. Since an entity can be made of several entities, then it is possible for us to use the inclusion label to show that. Since the function of an entity is also an entity, the inclusion label can also be used to show a function that includes other functions. For instance, if *Entity One* includes *Entity Two*, then the inclusion label can be used to show *Entity One* contains *Entity Two*.



Available Options

Available options for the inclusion label include

- Has/have
- Includes/include
- Contains/contain
- Inclusion entity
- Part of
- Exist with
- Given with or is given with
- Is a part of
- Contain
- Etc.

The Attach Label

The attach label or the attach entity can be used to show an entity that is attached to another entity. For instance, if an entity connects to another entity, then the attach label can be used to show that. We can position the attach label to our desired orientation.



Available Options

Available options for the attach label include

- Attach
- Connect
- Match entity
- Etc.

The Interaction Label

The interaction label can be used to show the interaction of an entity. For instance, we can use the interaction label to show the interaction of a person with an entity. Since information about an entity points to that entity, the interaction of an entity points to that entity as well. In this case, the interaction label can be used in conjunction with another entity to show the interaction of an entity with another entity.



Available Options

Available options for the interaction label include

- Interaction
- Interaction entity
- Interact

- Use
- Etc.

The Visually Identified

The visually identified label is used to show an entity that visually identifies another entity. For example, we can use the visually identifies label to show a word that visually identifies an entity.



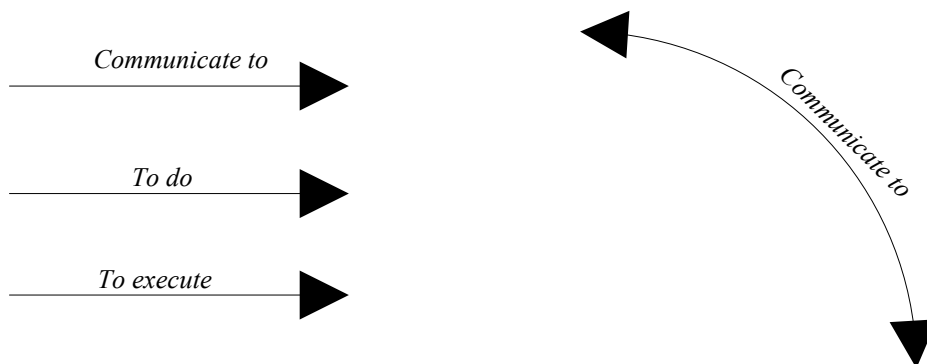
Available Options

Available options for the interaction label include

- Identify
- Point to
- Map to
- Match
- Shown by
- Etc.

Some Communication Related to Work Labels

The following labels can be used to construct the communication related to work model. We can annotate the labels with words that we like. For instance, the *communication to* can be replaced by *talk to*, *speak to* etc.



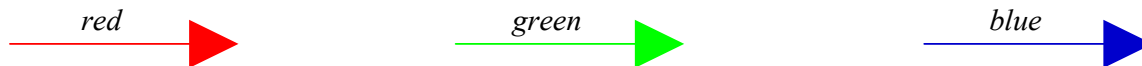
The Use Label

The use label is used to show an entity that uses another entity; for instance if *Entity One* uses *Entity Two*, then we can use the use label to show that. In this case, while *Entity One* uses *Entity*

Two, however *Entity One* is not always a part of *Entity Two*.



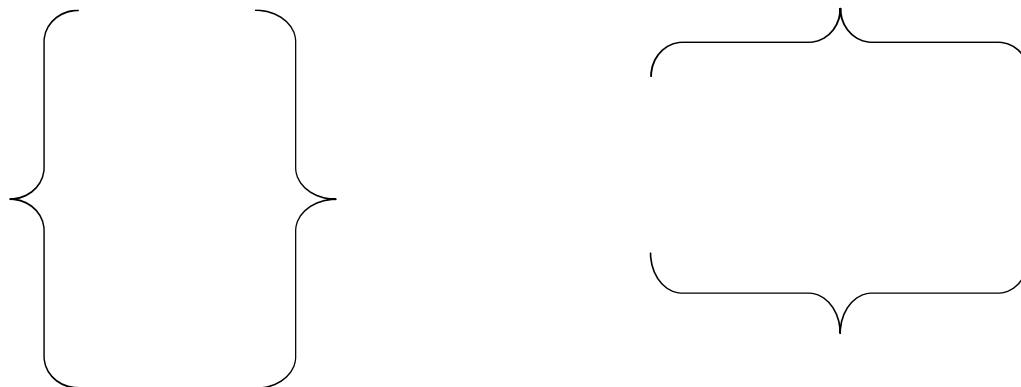
Communication Signals Direction



The communication signals direction arrows can be used to show the signal flow of our communication. For instance, the red arrow can be used to show the flow of a negative communication, the green a can be used to show the flow of a positive communication, while the blue arrow can be used to show the flow a communication that is considered as feedback.

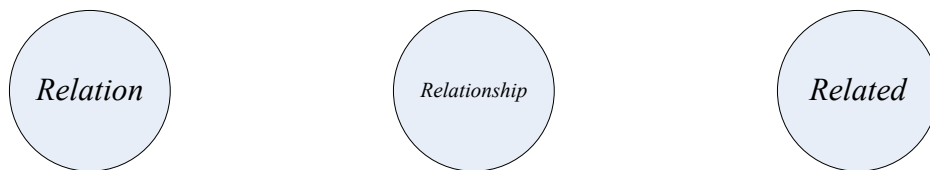
Curly Brackets

If necessary or needed, it may be possible for us to use curly brackets for annotation in our models. The curly brackets below can be used to provide more information with an entity.



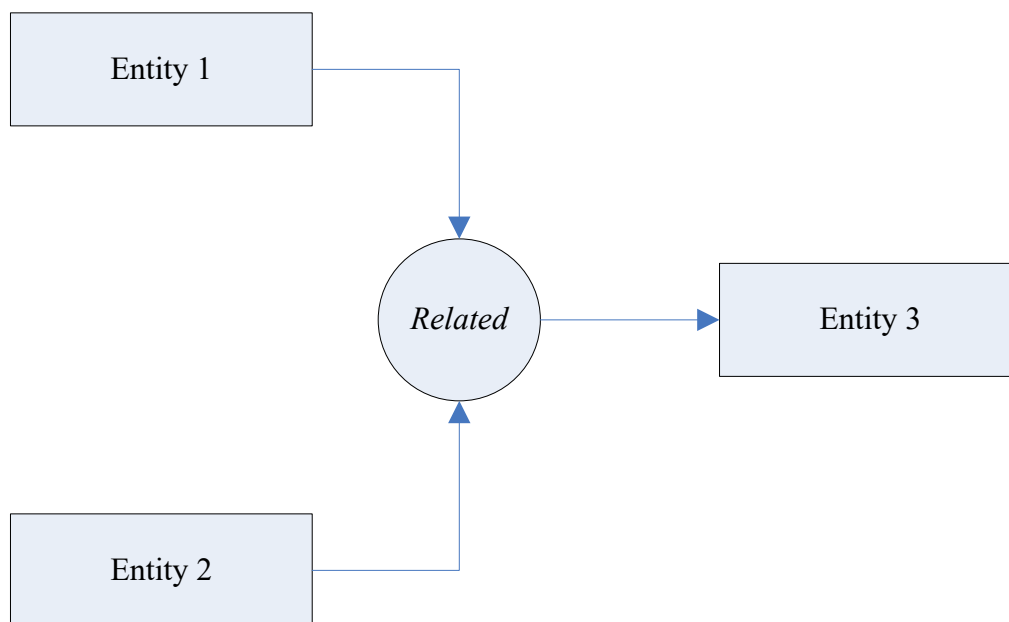
The Relationship Entity

While it may not be necessary, since the relationship between two entities is also an entity, the relationship entity below can be used to show that. The relationship entity is given in this form.

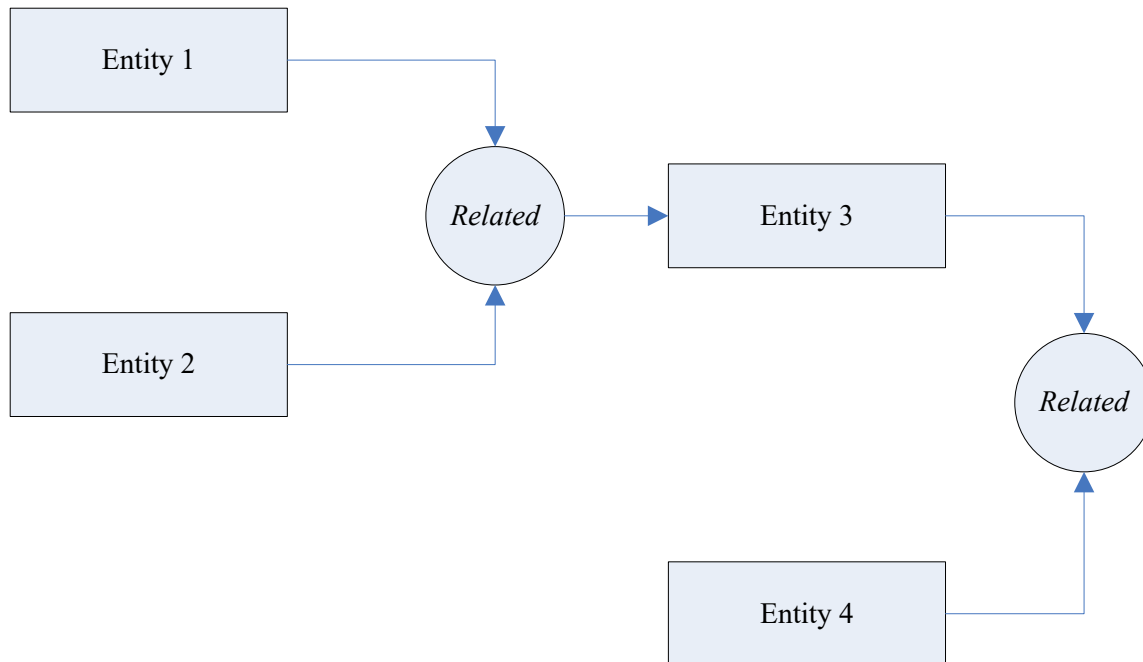


Usage Information

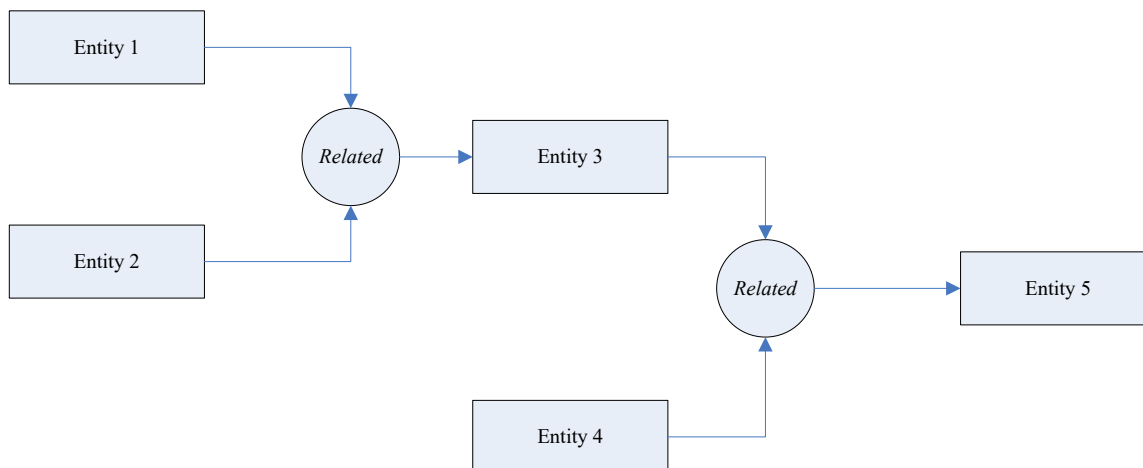
Assuming that entity 1 is related to entity 2 by entity 3, since the relationship of two entities is also entity, the relationship entity can be used in the form below to show that. We can say that entity 3 is the result of the relationship of entity 1 and entity 2.



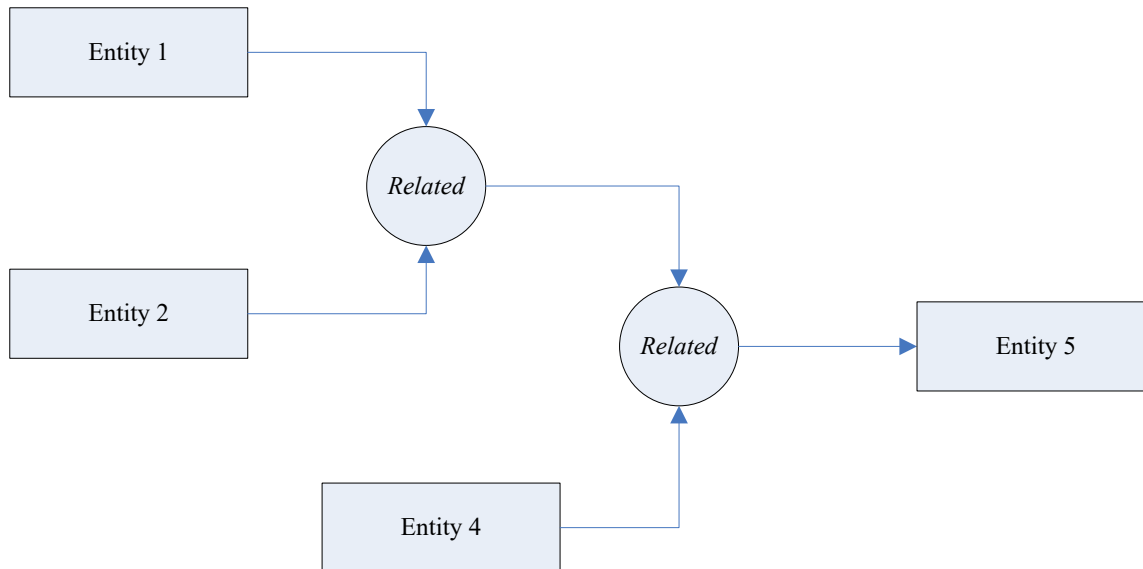
The relationship entity can also be used for multiple relationships in the form below. Assuming that there is also a relationship with entity 3 and entity 4, in this case we have



Now since the relationship between entity 3 and entity 4 is also an entity; let's assume that we name that entity, entity 5; then the model can be changed to



By understanding the relationship entity itself from the model above, then the same model can be reduced to the form below, where entity 3 is now shown, since it is already taken into consideration in the relationship.

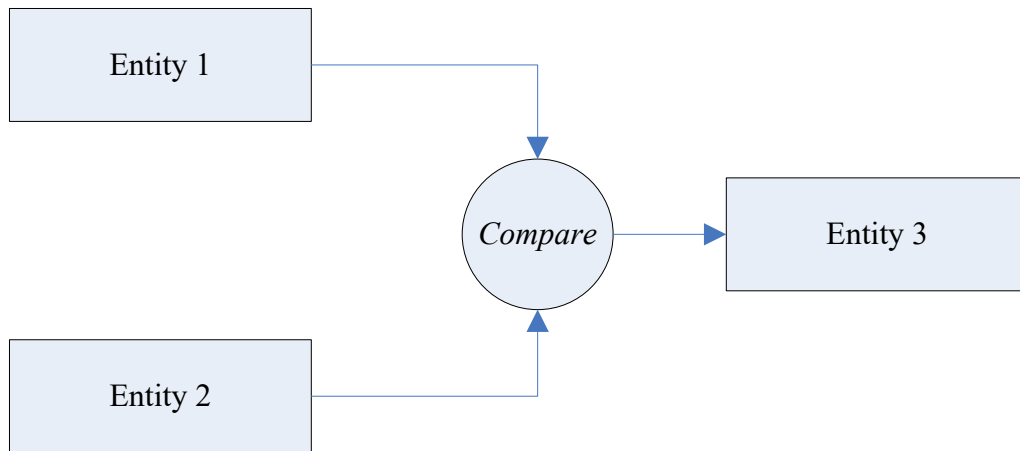


Available Options

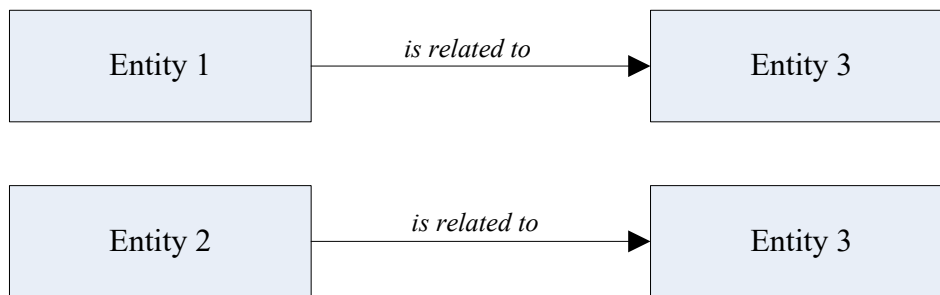
Available options for the relationship entity include:

- Relationship
- Relation
- Related
- Any relation name
- Difference
- Similarity
- Comparison, compare etc.
- Attached
- Etc.

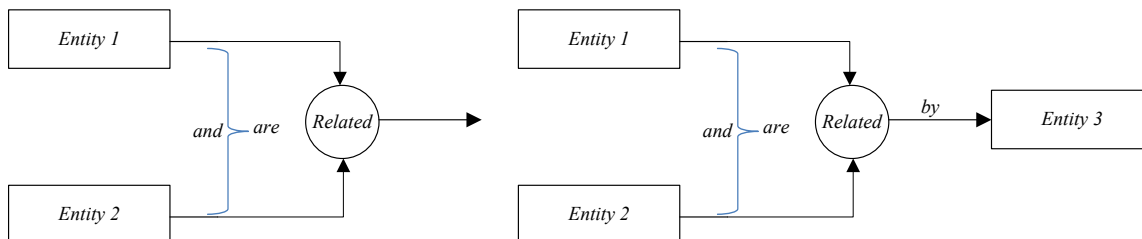
While the relationship entity can be used for comparison, nevertheless it is important to know that comparison of entities take understanding of the principle to a higher level. The way to look at it, since in order to compare two entities both of them must be well understood; in this case we simply increase our understanding of those entities. Since the understanding of communication depends on the principle itself, all that we do during the process is increasing our understanding of the principle. In term of comparison, the model below shows an example, assuming that entity 3 is the comparison of entity 1 and entity 2.

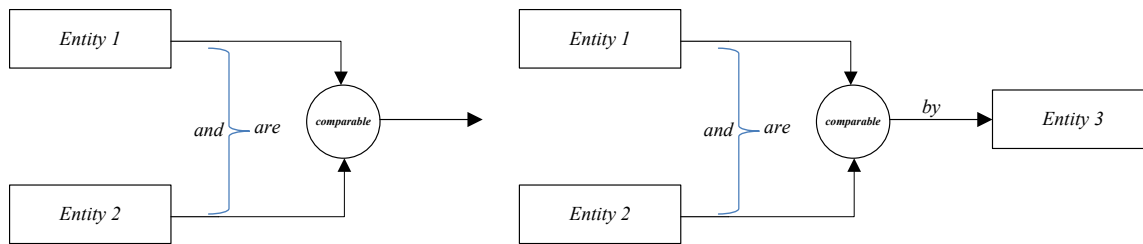


In term of relationship, the first model above shows that *entity 1* is related to *entity 3* and *entity 2* is related to *entity 3*. By using the relationship arrow, we can represent them as shown by the two models below.

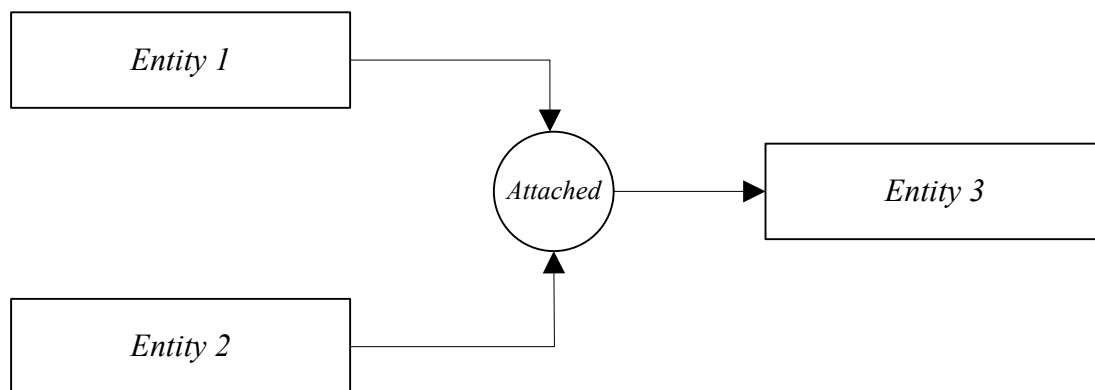


We can use curly brackets to annotate the relationship entity. By using curly brackets, we can annotate some of the models above in the form below. The first model below assumes that *entity 1* and *entity 2* are related, while the second one assumes that *entity 1* and *entity 2* are comparable.

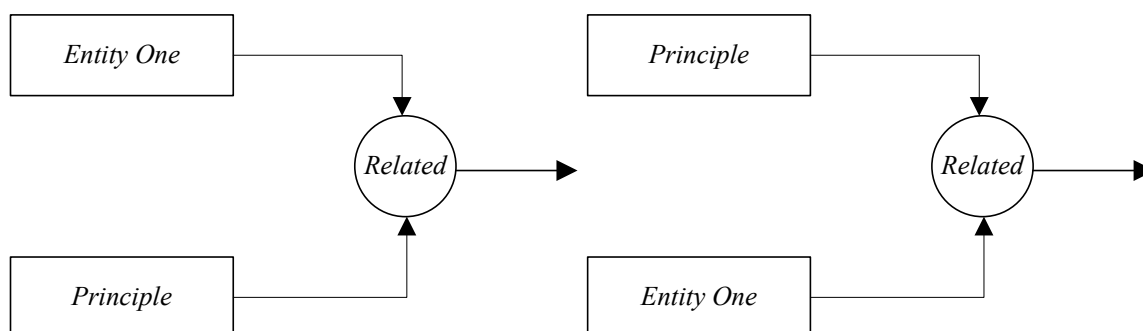




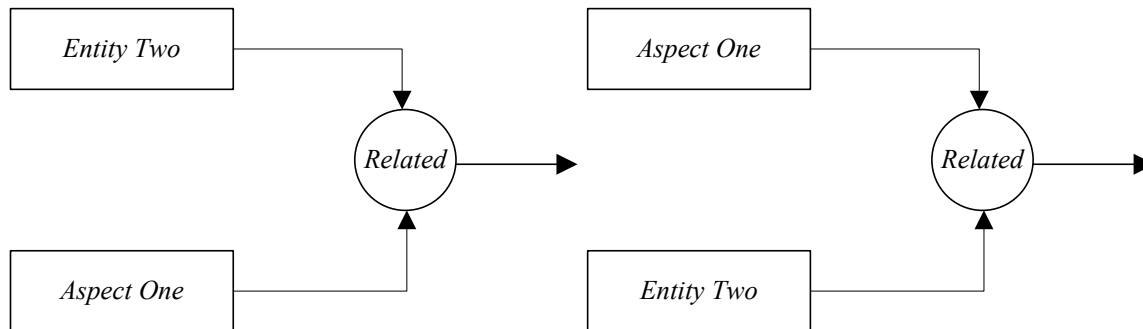
The relationship entity above can also be used as an attached entity to show an entity that is attached to another entity. Assuming that *Entity 1* is attached to *Entity 2* by *Entity 3*, then we can use the relationship entity with the attach option to show that. In this case, the relationship entity is used as an *Attached* entity as shown below.



Since we use the principle entity to validate other entities, the relationship entity can be used in conjunction with the principle entity to show the relationship between that entity and the principle entity as shown by the model below. Assuming that *Entity One* is related to a given principle, and then the relationship entity can be used with *Entity One* to relate *Entity One* with the principle entity.

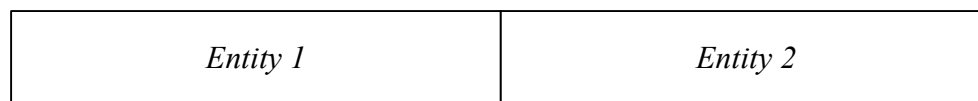


The relationship entity can also be used to relate an entity with the aspect of the principle entity as shown by the model below. Assuming that *Entity Two* is an entity, where *Aspect One* is an aspect of the principle entity, then if *Entity Two* is related to *Aspect One*, then we can related *Entity Two* to *Aspect One* by using the relationship entity.



Entity Attach to Other Entity

An entity that is attached to another entity can be show in a form represented by the model below. Assuming that *Entity 1* is attached to *Entity 2*, and then *Entity 1* and *Entity 2* can be shown by this model. The model above shows two entities that are attached by another entity, while the one below shows two attached entity. In the model above, we know the attached entity, while we do not know it from the one below.

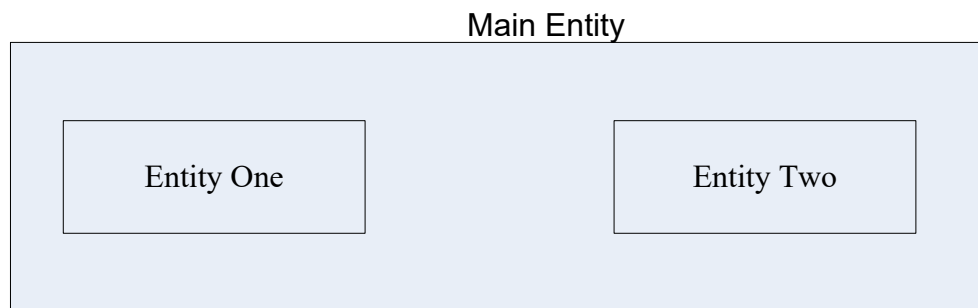


Identify Multiple Entities as an Entity

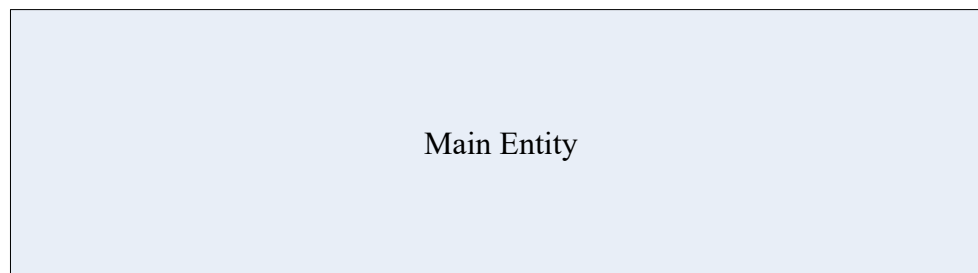
In term of entity, since an entity can be composed or included multiple entities, sometime it may be possible for us to show multiple entities as one entity. In this case, we can follow the instruction below to show that. Let's assume that we identify two entities: *Entity One* and *Entity Two*. In this case, it may be possible for us to show those two entities as one entity as shown below.



The two entities above are the entities that we identify. Assume that those two entities are part of another entity, in this case it is possible for us to show them as one entity as shown below.



The way to look at it, assuming that we have a main entity that makes up *Entity One* and *Entity Two*, in this case we can show both entities as one entity. That makes sense, since the main entity is also an entity, so do the parts of that entity. We can draw an arrow from *Entity One* to *Entity Two*; since the main entity is also an entity, in this case we can show the main entity below without being shown *Entity One* and *Entity Two*. While we call it main entity here, any other name can be used. The way to look at it, we can use the entity element to show entities as one entity.



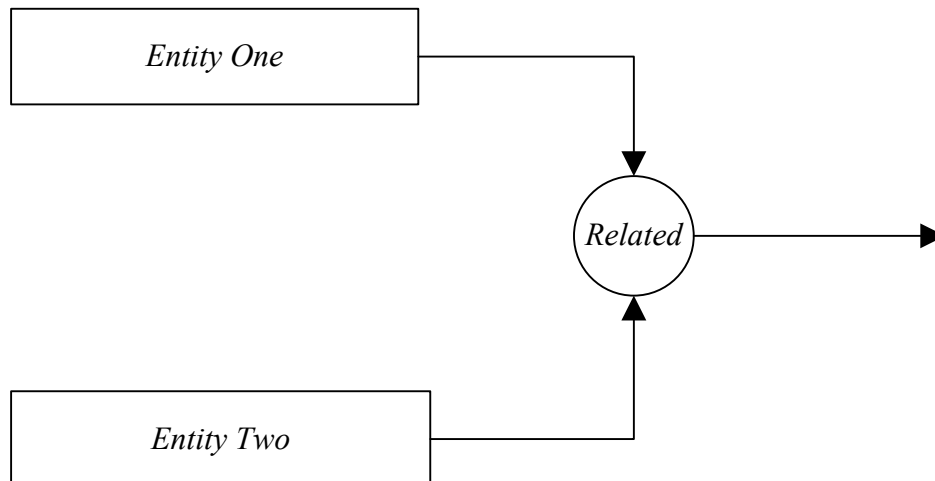
Identify Part of an Entity

Given that a part of an entity is also an entity; given that a section of an entity is also an entity, it is possible for us to use the entity element to show the identification of an entity that is a part of another entity. Assume that from *Entity One*, we identify *Entity Two* and *Entity Three*, where those two entities are part of *Entity One*. In this case, we can show or identify those entities in the form below. In this case, *Entity One* is considered to be the main entity that we identify.

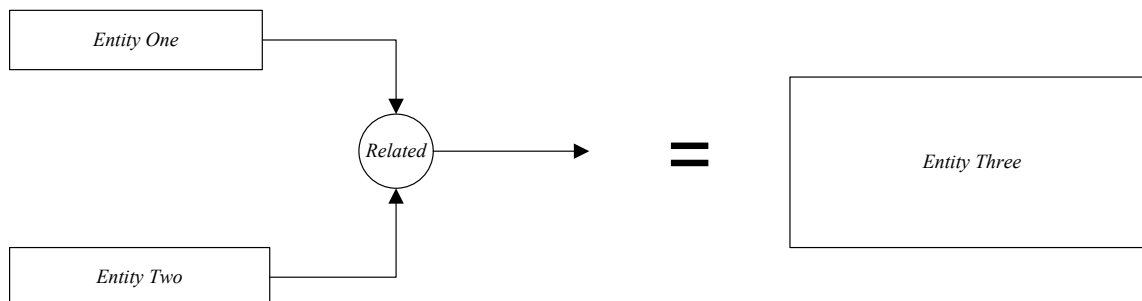


Identify Relationship as an Entity

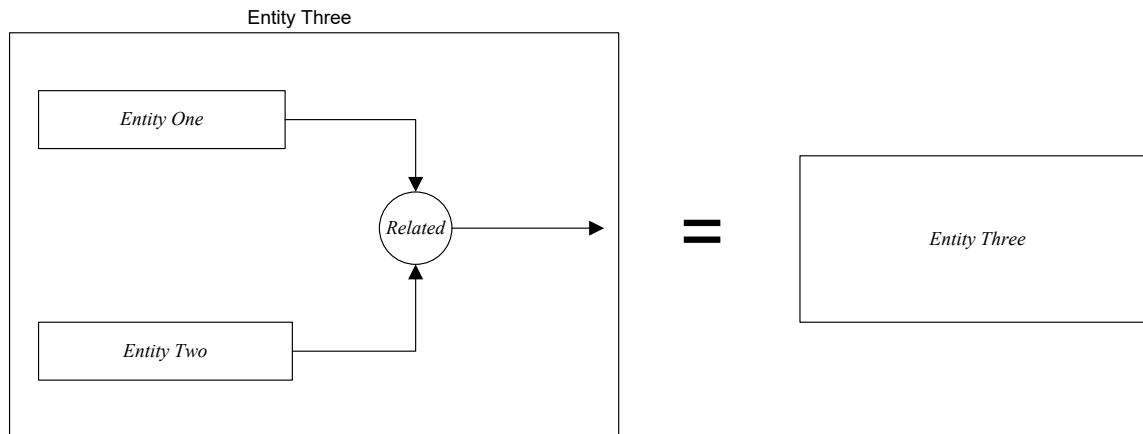
Since the relationship of an entity is also an entity, in this case it is possible for us to show or identify a given relationship as an entity. To better understand that, let's take a look of the model below. Assume that we have a given relationship as presented below.



Since the relationship between *Entity One* and *Entity Two* is also an entity, in this case we can show that entity in the form below as an entity.



Entity Three is being identified as the whole relationship of *Entity One* and *Entity Two*; to better understand that, we can represent it in the form below.



While we show it in the form above, since there is no limit in terms of numbers in a relationship, there is no limit as well in terms of numbers of relationships that can be identified as an entity. Any relationship disregarding any number can be identified as an entity.

From the model above, the relationship entity inside the dashed area is being used for illustration only. While the relationship of an entity is an entity, it is always good for us not to show a relationship inside an entity. Since the relationship of an entity is also an entity, once we identify that entity, it is always better for us to show that entity as part of the main entity, rather than showing the relationship inside the main entity.

The way to look at it, while we can identify a relationship inside an entity, but we cannot show a relationship inside an entity; since that relationship is being viewed as an entity, it is better for us to show it as an entity. If we were going to show that entity as a relationship, inside another entity, that will further increase the complexity of our model and we will also lose our objective or focus. Our objective is to reduce complexity to help us understand and solve problems not to increase complexity. Keep in mind that does not exist on a piece of paper or a computer screen.

The Location Entity



Usage and Description

The location entity can be used to identify a location or a site of operation. Given that the functions that we execute do not take locations into consideration, while we can use the location entity to identify a location, nevertheless it is not necessary for us in the modeling of our application. Since our application does not take location into consideration, it may not be necessary for us to use the location entity to model our application.

While the modeling of our application does not take location into consideration and should not take location into consideration, nevertheless if there is a need to identify the location or the site that we operate within our application, the location entity can be used to identify that site or location. For more information about using the location entity, refer to the example section.

Available Options

Available options for the location entity include:

- Location
- Site
- Location with index or number
- Site with index or number
- Location of operation
- Site of operation
- Etc.

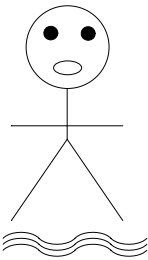
The Mobility Entity



Usage and Description

Use the mobility entity to show the mobility of a person. Given that we are mobile we can move from location to location. The mobility entity can be used with the person entity to show the mobility of a person. Below is a usage example that shows that Person 1 moves from location 1 to location 2.

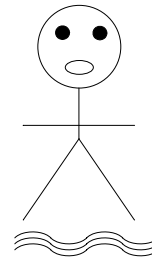
Person 1 at location 1



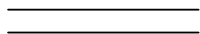
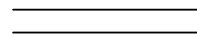
moves to



Location 2

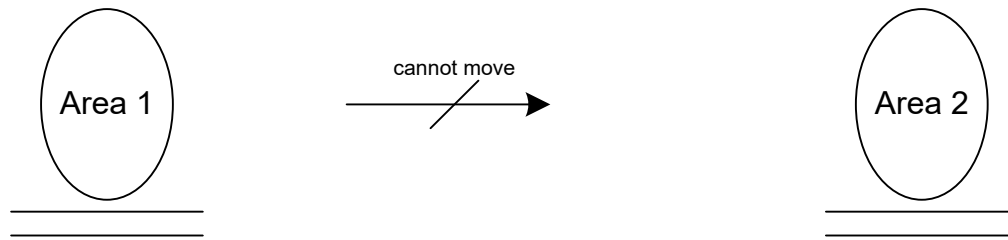


The Fixed Entity

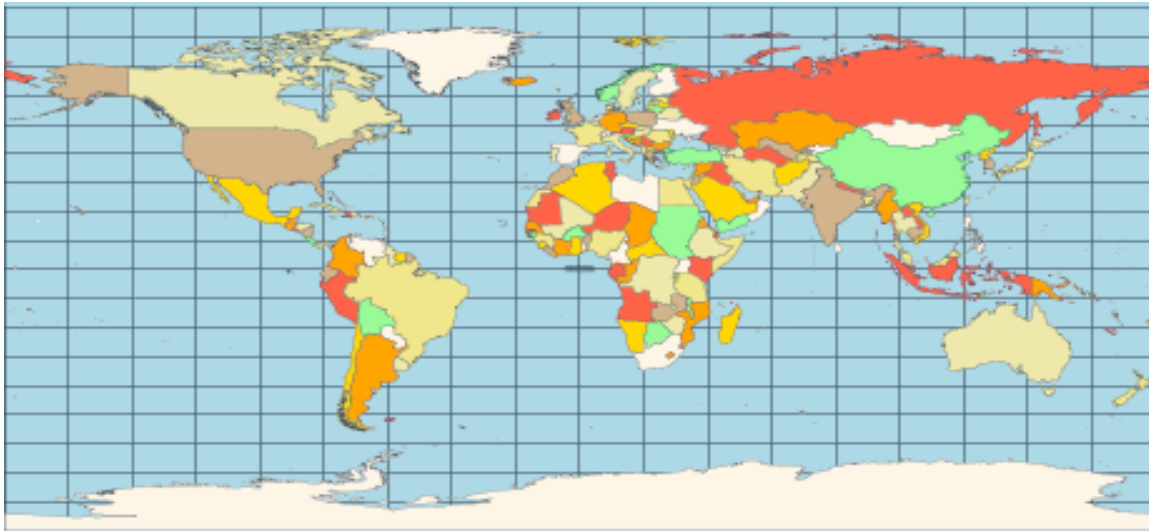


Usage and Description

The fixed entity or immobility entity is used to represent an entity that is not mobile or cannot be moved. For example, we can move the immobility entity with the area entity to show that the area entity cannot be moved. The example below shows that Area 1 cannot move to Area 2, because both Areas are immobile.



The Area Entity



Area

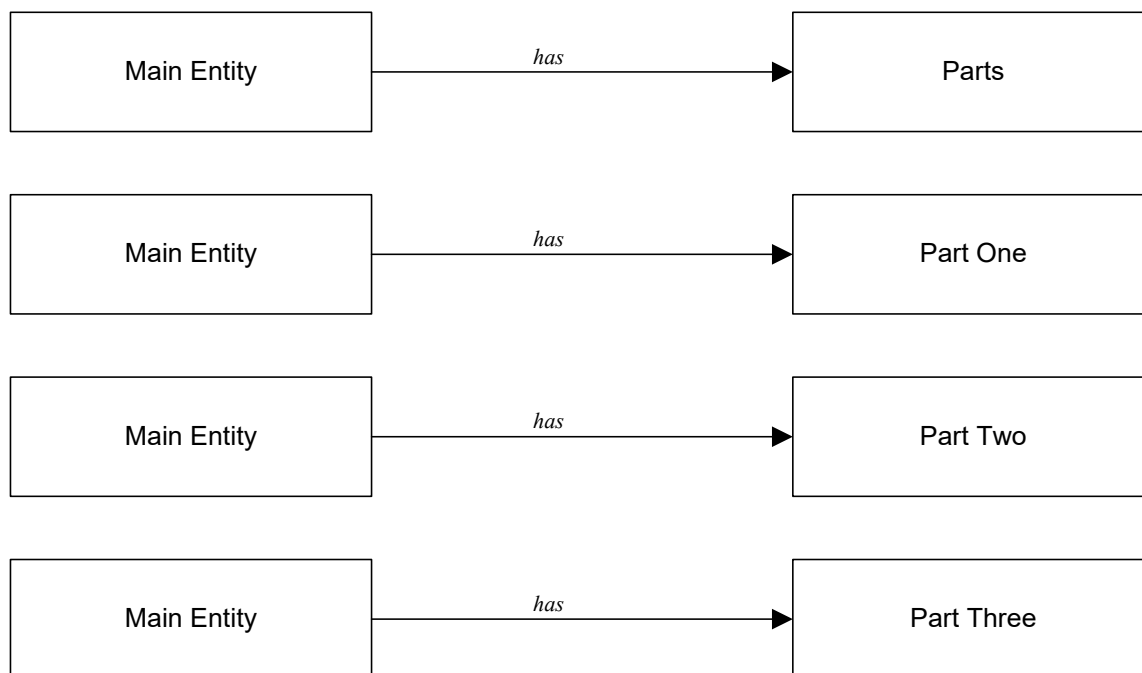
Usage and Description

We can use the area entity to show our area of our operation. Let's assume that we operate in specific country or area, then we can use the area entity to identify that country as our area of operation. The area entity can also be used with the site of operation entity to show both our site and our location of operation. For more information about using the area entity and the site

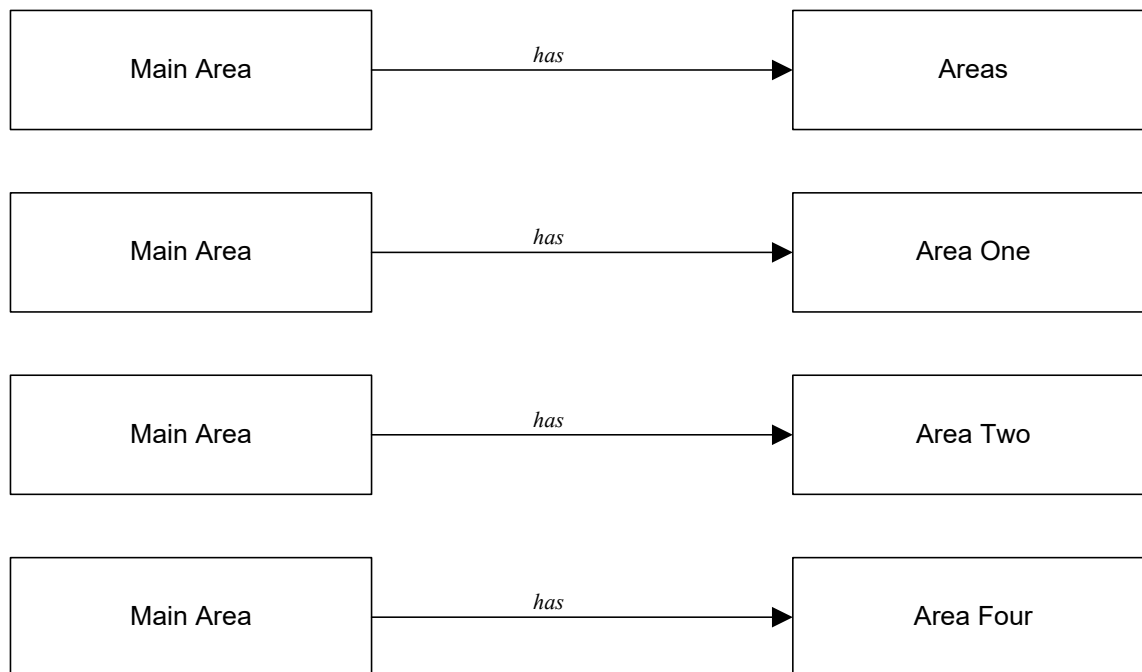
entity together with the location entity, refer to the example section. By understanding entity and parts of entity, the area entity can be viewed in the form below as a main entity.

Each area or country in the main entity can be viewed as a part of that entity. For instance, if we identify *Country One*, then *Country One* is viewed as a part of the main entity or simply as an entity within the main entity. In this case, we can label them as parts of the main entity as shown below.

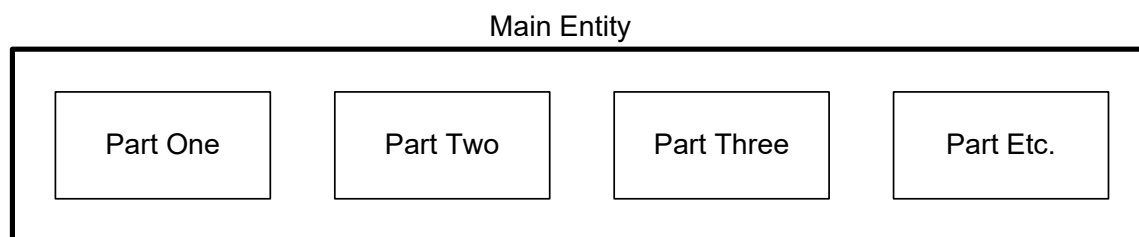
To better understand the area entity, it is always good for us to view it as the main entity. In this case, we can view each area in the main entity as part of that entity. For instance we can think the main entity in the form of the model below, where the main entity has parts. In this case, if we identify *Part One*, *Part Two*, *Part Three* and so forth. those parts are considered as parts of the main entity, where each part represent an area of the main entity. This is the same as saying the main area has *Area One*, *Area Two*, *Area Three* etc.



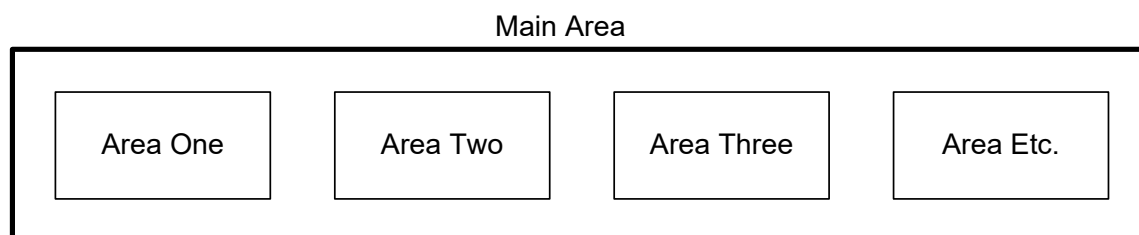
As we said above, the main area includes areas, where those areas are considered to be areas of the main entity. For instance if we identify *Area One*, *Area Two*, and *Area Three*, those areas are considered to be parts of the main entity. In this case, we have



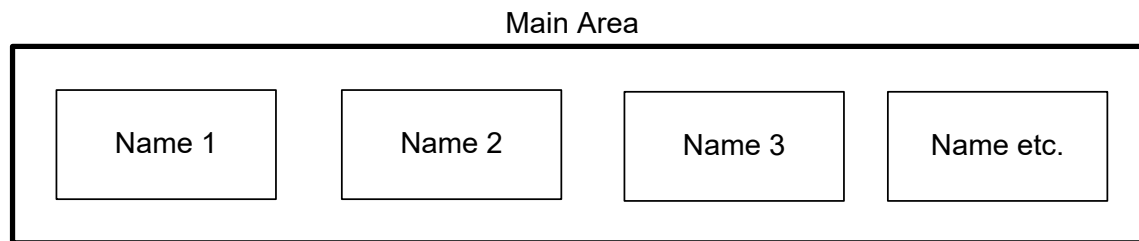
By understanding the explanation and the models above, we can represent the main entity in the form where we can show the main entity and the parts of the entities that include inside that entity. In this case, we have



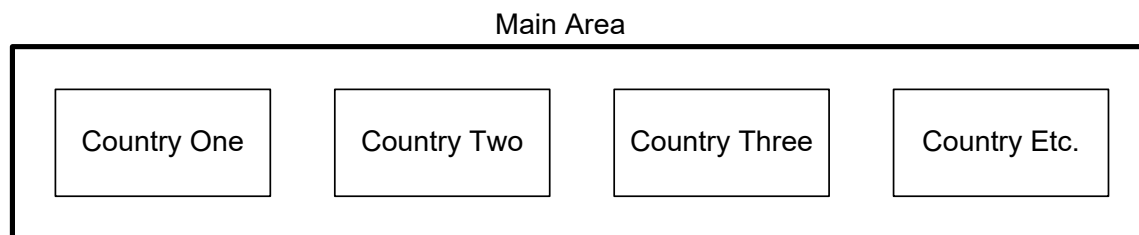
From the model above, each part of the main entity is considered to be an area, a country, a location etc. The model below represents the main area and the areas that include inside the main area.



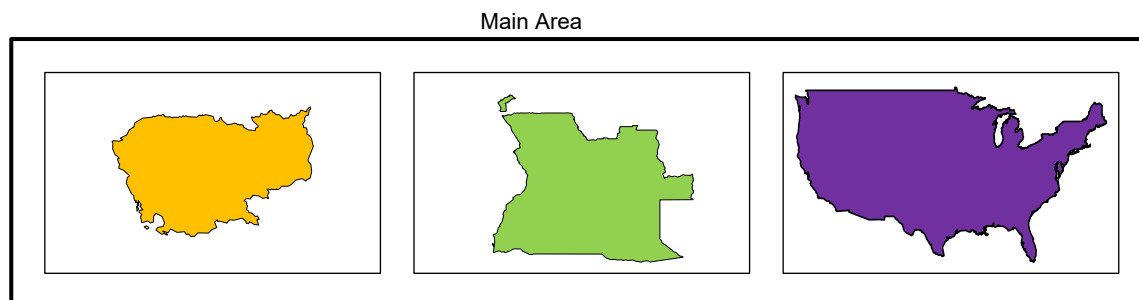
From the model above, if we identify the name of an area or the name of a part of that entity, we can also show that in the main entity in the form below. In this case, *Name 1*, *Name 2*, and *Name 3* are viewed as the name of each part of entity that we identify in the main entity.



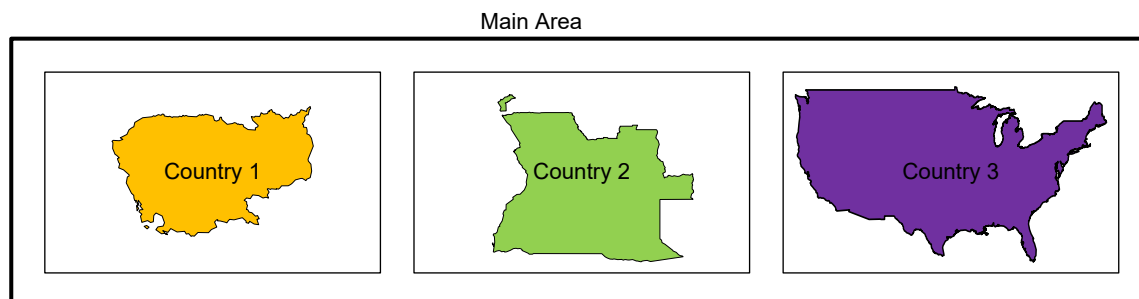
Although that may not be important, but if we want to, it may be possible to show the names of countries or the image or sketch area represented by countries



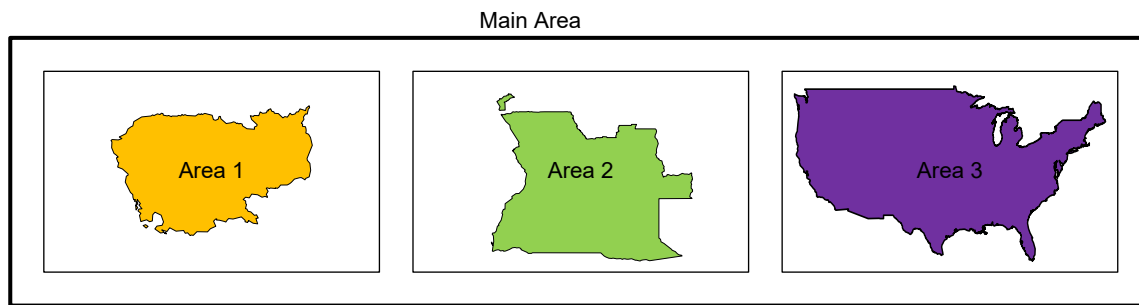
The model below show the sketch of each area or country represented in the main area entity.



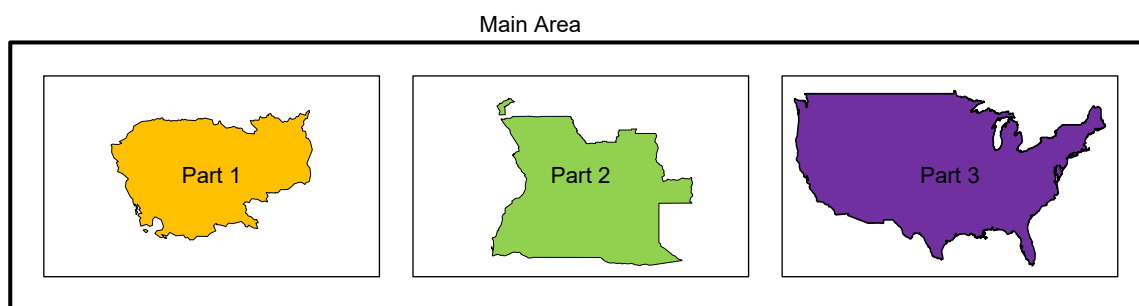
In the model below, we represent the countries as *Country 1*, *Country 2*, and *Country 3*. We can also use the name of the countries to represent each area that makes up the main area entity.



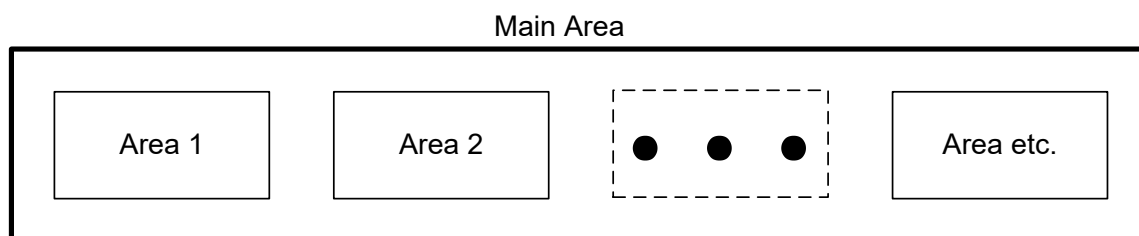
Below, we label them as *Area 1*, *Area 2*, and *Area 3*



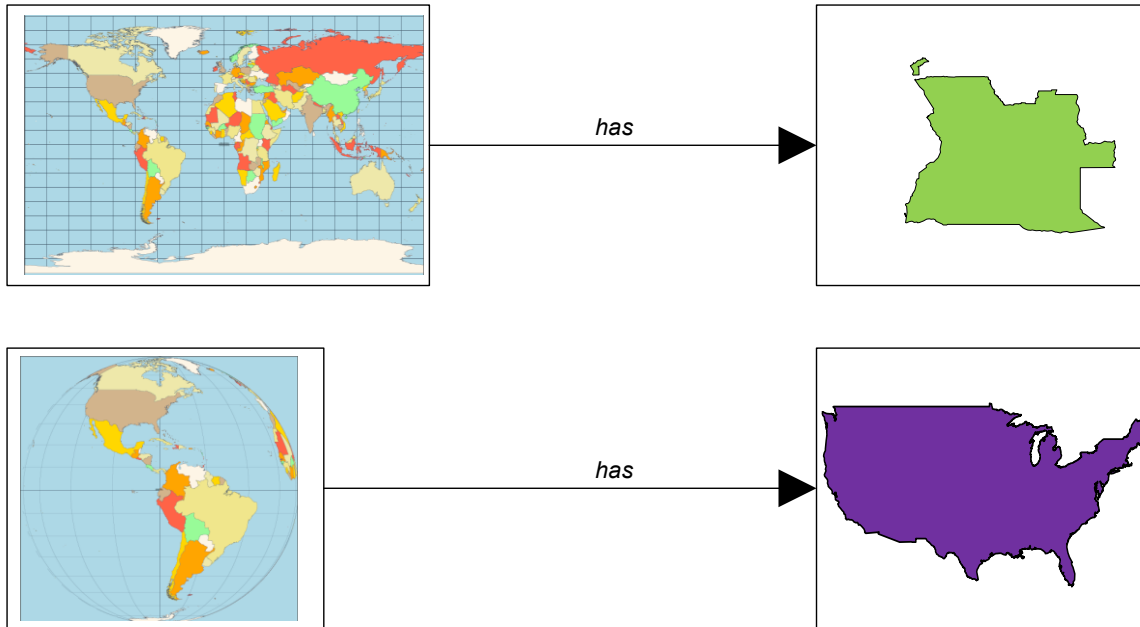
Since each area is considered to be a part of the main entity, in the model below we label them as *Part 1*, *Part 2*, and *Part 3*.



The main area can also be used with continuity in the form below. In this case, we use the continuity to show more areas that are not visible in the model below.



In term of entity and parts of entity, we can also show the main area entity in the form below, where we can use the *has* relationship label to show an area or a country that is a part of the main entity. In this case we have

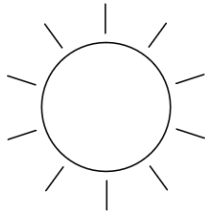
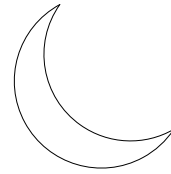


Available Options

Available options for the area entity include:

- Area
- Area of operation
- Area with index or number
- Location
- Location with index or number
- Country
- Country with index or number
- Name with index
- Country name
- State
- City
- City with index or number
- Place
- Etc.

The Natural Observer Entity



Usage and Description

While it may not be necessary, however if needed the natural observer entity can be used to show the observation of a model or a communication or action or function execution and so forth. When using the natural observer, it can be put to the upper right corner of a model or a sheet of paper or a page to show the natural observation.

The way to look at it, a model is natural, the natural observer is used to show that the model is natural. The function of a communication is natural, the natural observer is used to show that. A communication is natural, the natural observer is used to show that. An entity is identified, the identification of that entity is natural, the natural observer is used to show that.

Available Options

Available options for the natural observer entity include:

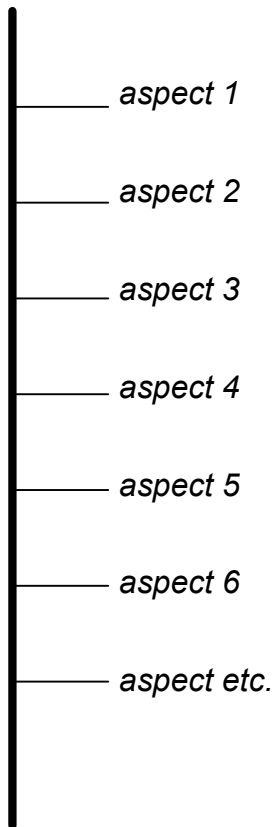
- Natural observer
- Visual observer
- Mother nature
- Parent
- Sun
- Moon
- Cloud
- Natural entity
- Natural element
- Etc.

Entity and Aspect of Entity Model

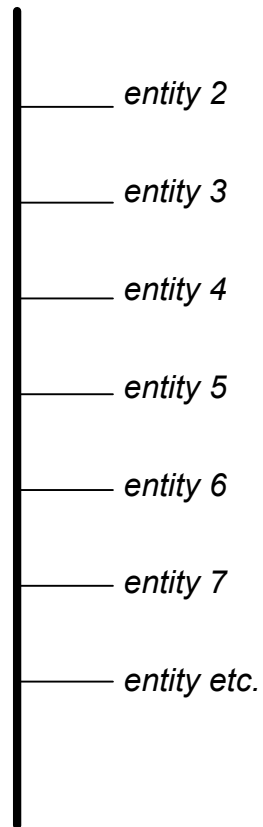
Entity and aspect of entity model can be shown in the form below. For instance, if we have an entity that has several aspects, then we can use the form below to show that if we want to.

Assume that entity 1 has several aspects. In this case, entity 1 has aspect 1, aspect 2, aspect 3, etc. Then we can show that by using the models below. Since the aspect of an entity is also an entity, the model to the right shows that. In this case, we use the models below to show an entity that includes other entities.

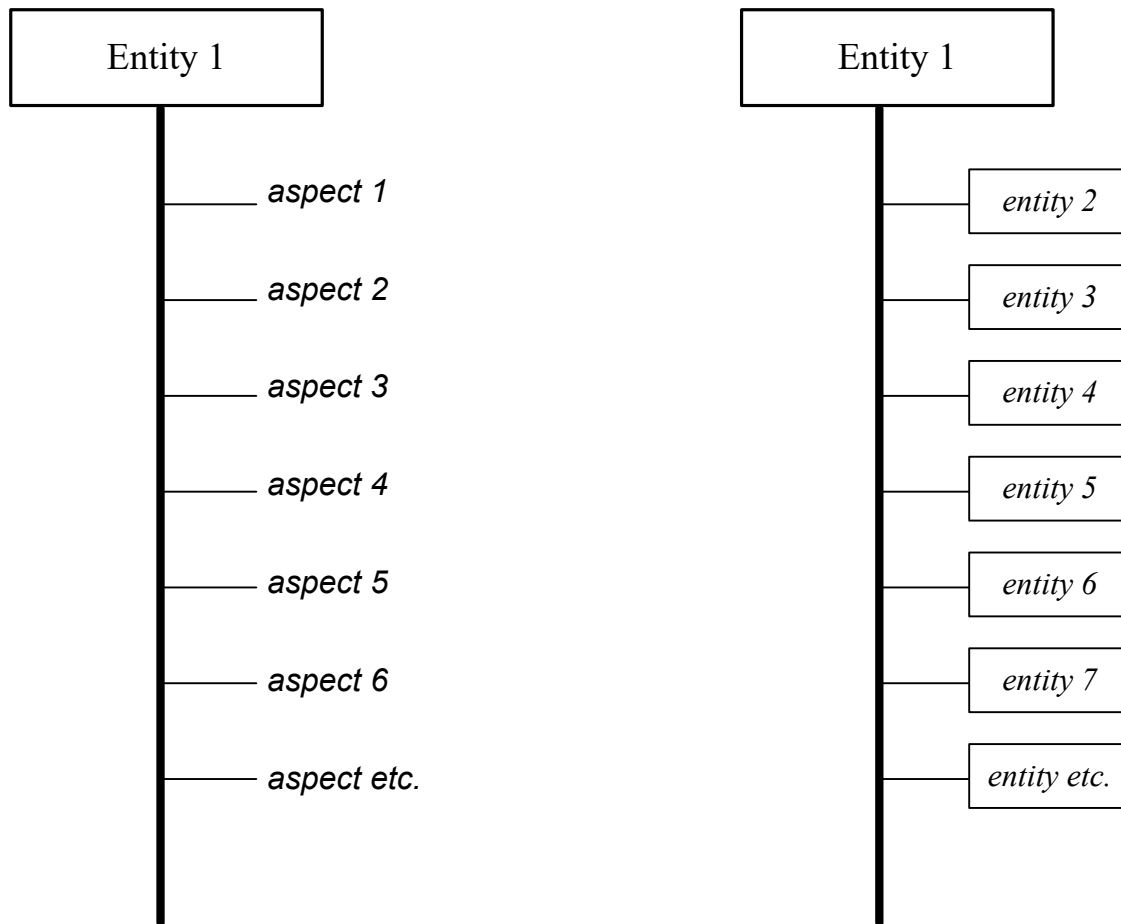
Entity 1



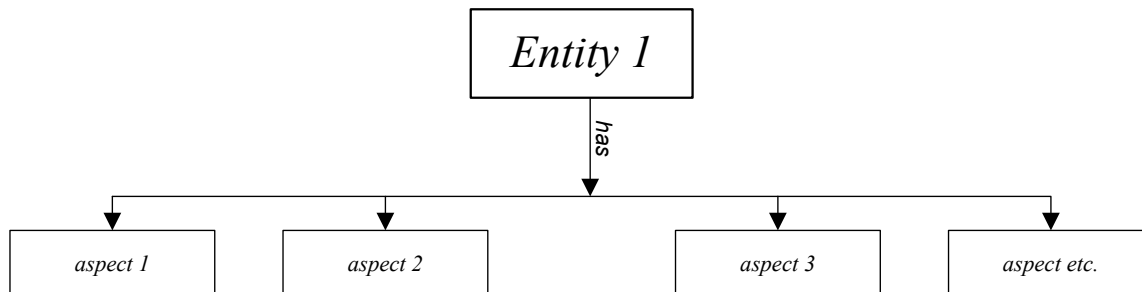
Entity 1



The two models below are the same as the ones above



The same model above can be shown in the form below. Here we simply use another form to show it. From the model below, we only show 4 aspects. The other ones are not listed.



Change of Communication Function from Communication Tabulated Form

Communication	Application Value	Communication Function

Usage and Description

Since our application is communication driven, it makes sense for us to show how any change in our communication affects the application. Since our project is communication enabled, any change in our communication affects the project. By using the table above, we can provide information about how our project change related to communication.

Consider the overall process as our application, we can also say that the change in our application related to communication. We use the change of communication function from communication table to show how our application execution changes from communication. While the table above shows the basic process, it is always good to expand it to show more information. The following tables can be used to show more information about the change of our application from communication. Keep in mind that, they are the same as the one above, but extend it by providing more information.

Communication	Application Value	Communication Function	Communication Name

Communication Order	Communication Subject	Communication Execution Function	Application Value

Communication Order	Communication Subject	Communication Execution Function	Application Value	Application Name

Communication Order	Application Value	Communication	Application	Communication Function

Communication	Application Value	Communication Function	Communication Name

Communication Order	Communication Date & Time	Communication	Application Value	Communication Function

Communication Order	Communication Date	Communication Time	Communication	Communication Result

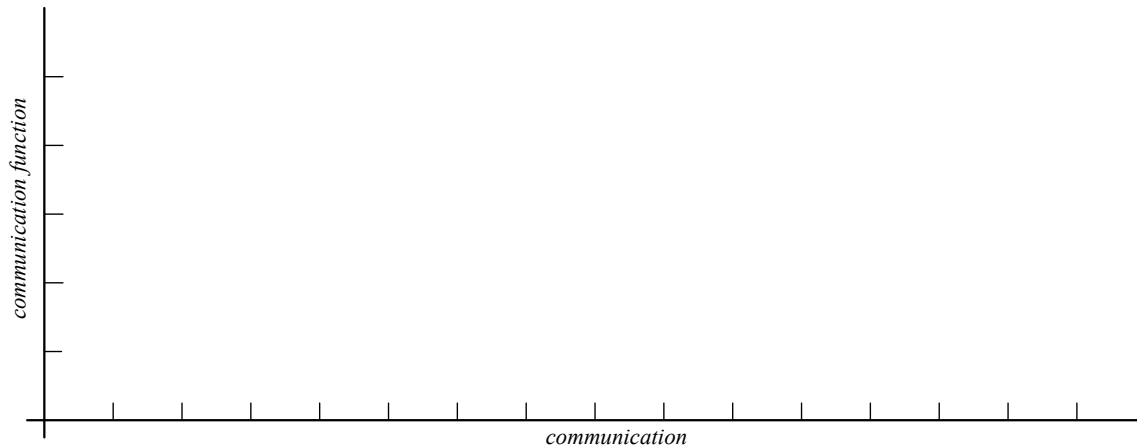
Available Options

Available names for the change of communication function related to communication include

- Change of application related to communication
- Change of application from communication
- Change of communication function related to communication
- Change of communication result related to communication
- Change of communication execution function related to communication
- Change of application function related to communication

- Change of application execution function related to communication
- Change of application result related to communication
- Etc.

Change of Communication Function from Communication Graphical Form



Usage and Description

While we can use a table to show the change of application from communication, we also use the graph above to show or plot how our communication function changes related to our communication. For instance, if we are working in a project and we consider that project as our application, since that project is a function of our communication, we can use the graph above to show how that project changes from our communication. Since we consider that project as our application, we can use the graph above to show how that application changes from communication.

While we can use multiple tables as shown above in the table version to provide more information about the changes, we can also use the graph to show the same information as well. In this case, we can customize the graph axis and the labels to show the information. We use the graph axis above to show or plot the change of the communication function related to communication.

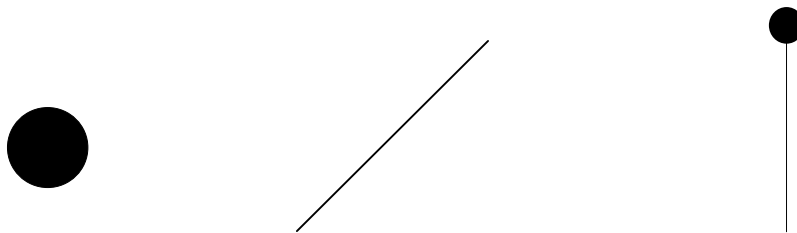
Available Options

Available options name for the graph above include

- Change of communication function
- Change of application

- Change of communication execution function
- Change of application result
- Change of the project
- Etc.

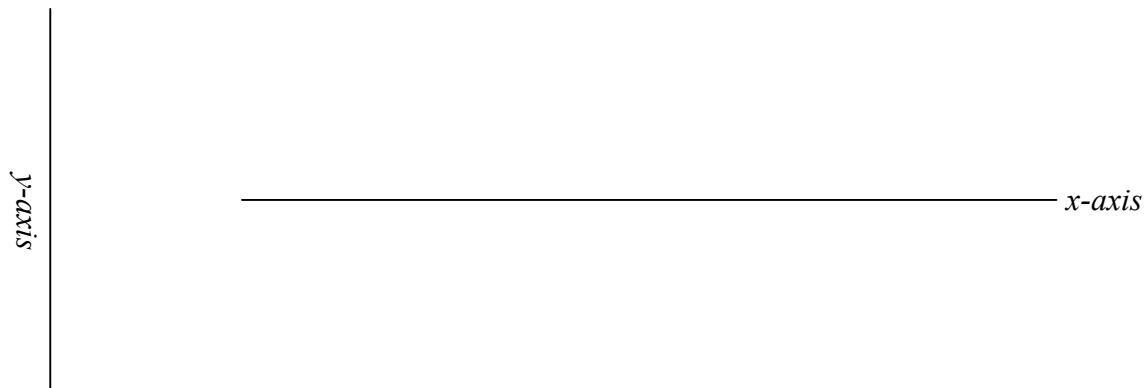
Graph Line and Graph Point



Description and Usage

When using the graphical form to show the change of our application related to communication, we can use the graph line and point with the axis to show the changes. We can rotate the line to any location to reflect the changes. We can also extend the line and add to it when constructing our graph.

Graphical Axis Entity



Description and Usage

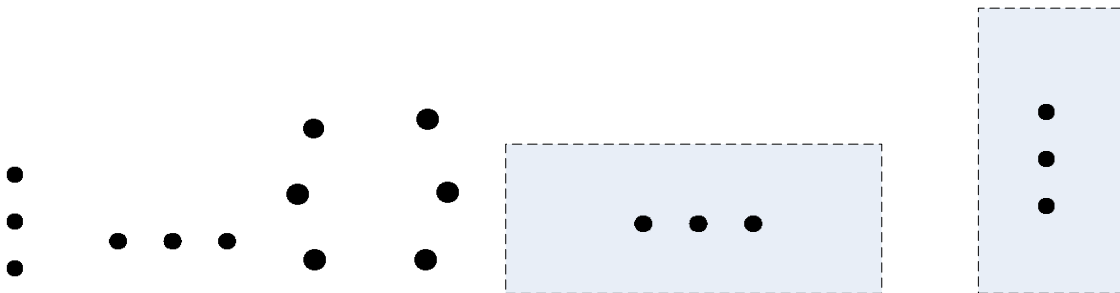
Depending on the information we want to show, we can use the graphical axis entity above for that. The graphical axis entity can be used together with graphical lines and points to represent information. Since graphs are considered to be communication entity, then we can use the

graphical axis with graphical lines and points as communication elements in term of communication to represent information.

Available Options

Available option for the graph axis entity includes any other name we want to give the axes.

The Continuity Entity

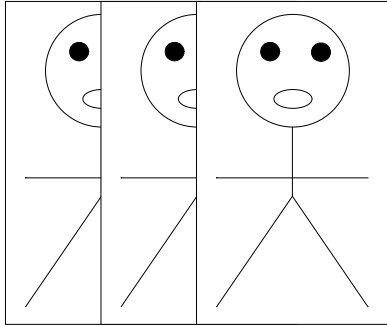
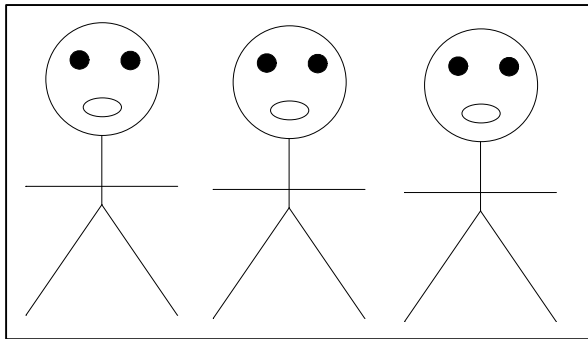


Description and Usage

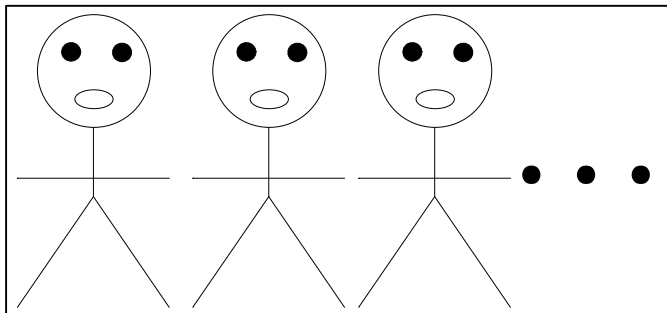
We can use the continuity entity to show the presence of continuity. For instance if an entity is to continue, we can use continuity to show that. For example, in a group of 50 people, we can show 3, and then use continuity at the end for the rest. We can also use continuity in the middle, and then show the last one at the end. Continuity can be used wherever it is needed. Continuity can be used with any entity, whenever it is needed. In other words we can use it for part of the communication function, part of application and with all others whenever it is needed. The arc type continuity can be used for the people work together model. Refer to the example section for more information about using continuity.

Group of People

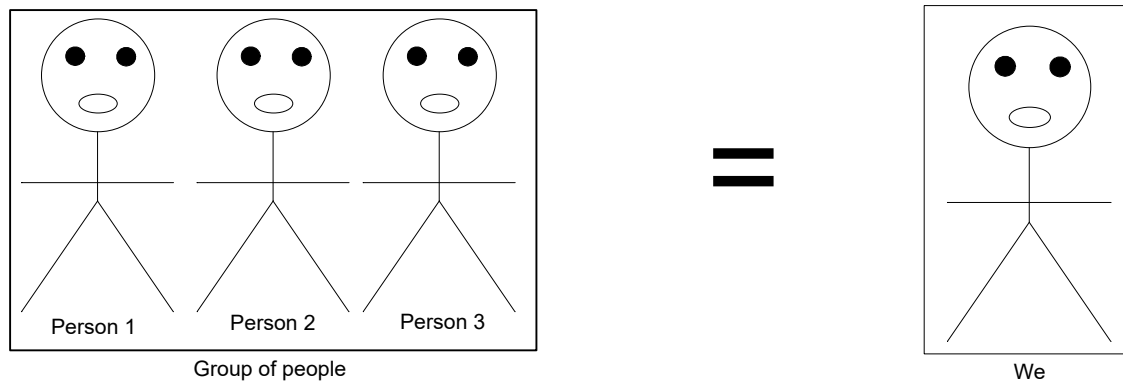
Within our project model, if space is an issue we can use the following models to show a selected group of people, see the example section for more information. Groups of people can be shown in the project by using both of the models below. We can use the first model to show a selected group of people on a computer screen, while the second one is to be used on a piece of paper or a drawing board.



When using the linear form of grouping as shown above, we can also use continuity to show a large group of people for example. The model below shows a group of 50 people. We use continuity to show that there are more people in the group than the ones that are visible to us as shown by the model below. We can put the continuity after two or after three it does not matter. The using of the continuity shows there are more people in the group.

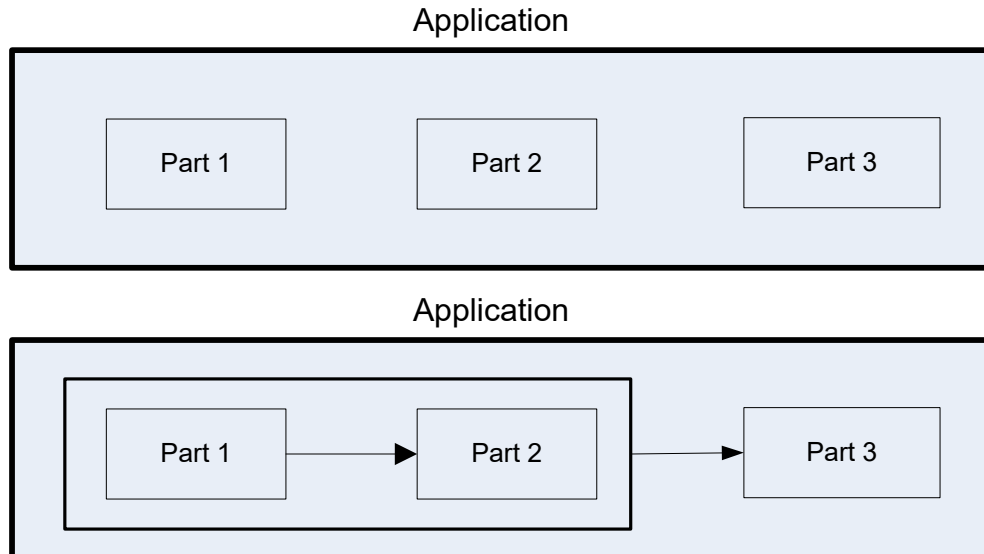


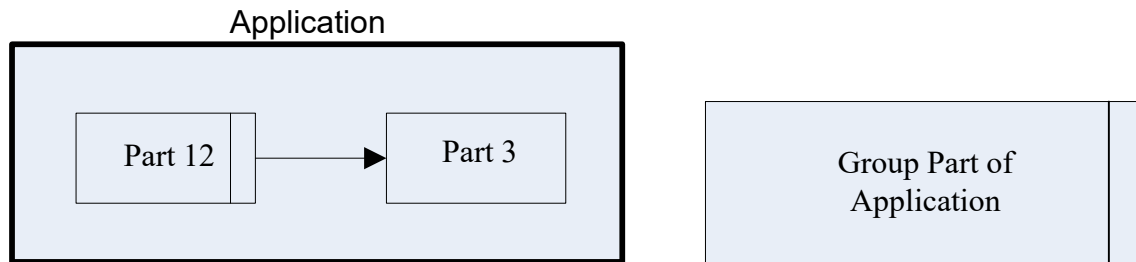
By understanding the person entity, the principle entity, and the relationship between the person entity and the principle entity, a group of people can also be identified as shown by the model below.



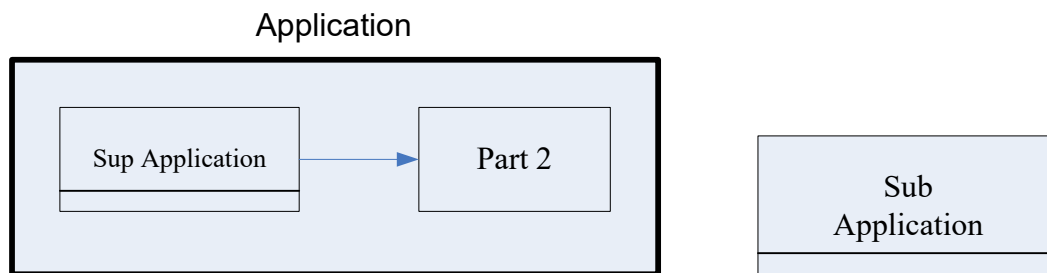
Sub Application and Part of Application

If the application is segmented into several parts, some parts of the application can be grouped to give a single part as shown from the models below. The first model shows that the application is divided into part 1, part 2, and part 3. In the second model, we simply use the empty container to group part 1 and part 2 of the application. To preserve space, we then use the compressed form of grouping to group part 1 and part 2, which results to the third model.





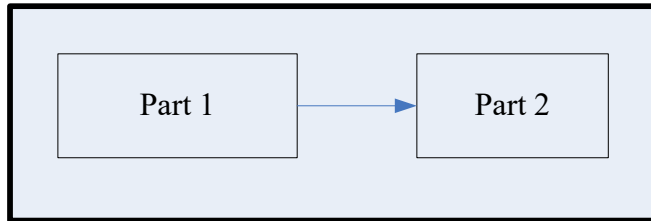
If the project is divided into several sections, it may be possible to have sub applications on that project; a sub application is considered to be a part of that application that includes other parts. A sub application can also be connected to a sub function. We can also say that the parts of a sub application can be linked to a sub function. The symbol of a sub application is similar to the one of a sub function. The model below shows the usage of sub application. From the model, we can see that the application is made of two parts, but the first part is considered to be a sub application. The model to the right shows the general form of a sub application.



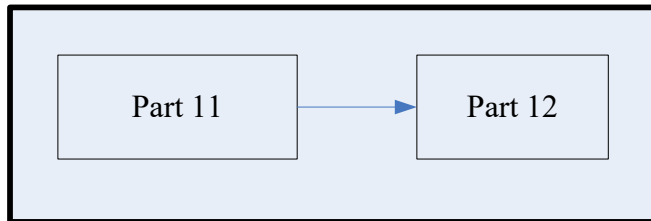
Sub Application Result and Part of Application Result

If the project is divided into several sections, it may be possible for us to use sub application result as well; a sub application result is considered to be a part of the overall result. Assume that our project is made of 5 sections, where each section is managed by a different person and different people work on each section. In this case, the result of the project will be the combined result of the 5 sections. To show that, we can use sub result or part of the result. The first model below shows that the application result is made of two parts: part 1 and part 2. The second model shows that part 1 of the application result includes other parts inside. In this case, part 1 of the application result is considered to be a sub application result. The identification of a sub application result is shown to the right of the third model and it is similar to a sub function. Parts of the application result can also be grouped similarly to parts of function by using empty containers.

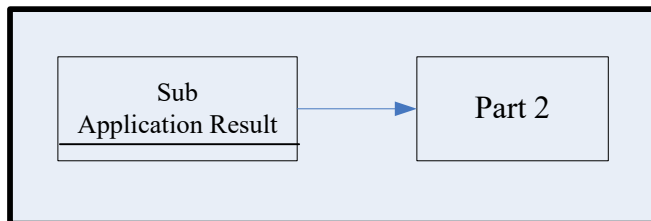
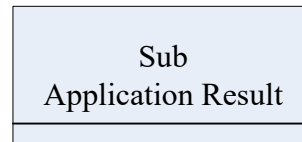
Application Result



Application Result Part 1



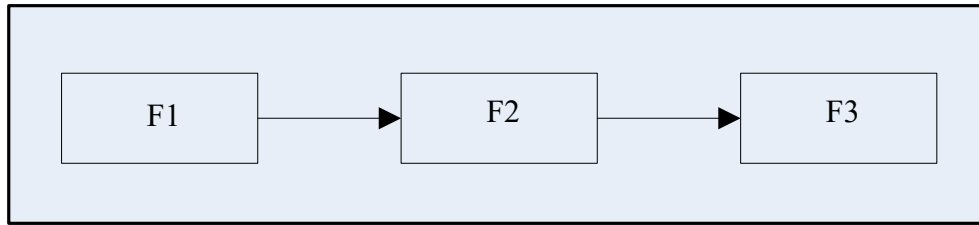
Application Result Part 1



Grouping Part of Communication Function

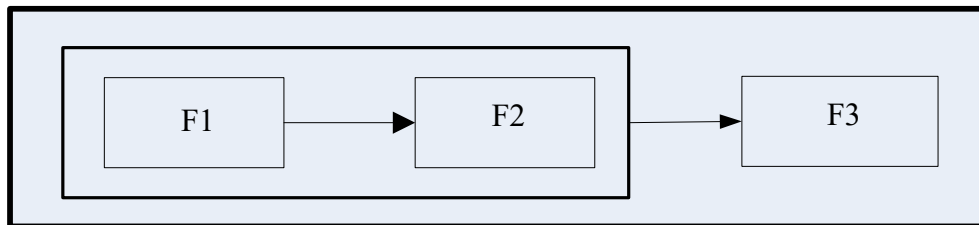
Since the overall project is driven by our communication, when managed by different people, it makes sense for the communication function to be divided into several sections. We have already seen that from the usage of the empty container entity. The model below shows that our communication function is made of three functions.

Communication Function



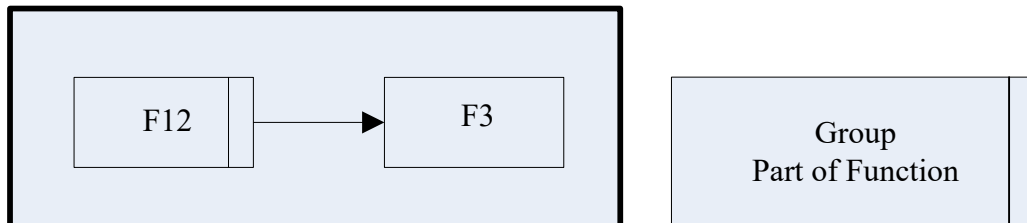
By using the empty container entity, we can group the first two functions of the communication function and leave the last one ungrouped as shown below.

Communication Function



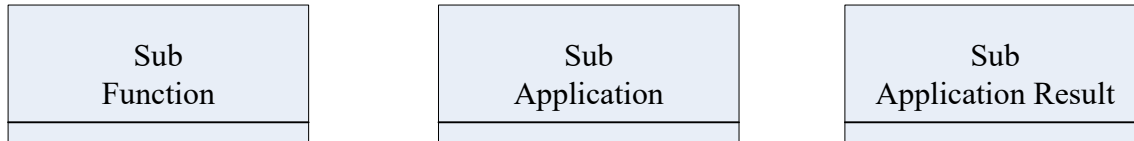
To reduce spacing, we then use the compressed form to group the two functions from the main communication function as shown here. The model to the right represents the compressed form of grouping parts of function. F12 is the combined grouping for function F1 and function F2.

Communication Function



Review of Grouping

In this section, let's review what we have learned about sub function, sub application, sub result, part of function, part of application and part of result. We use sub functions, sub applications, and sub results to link to other sections of our project. That usually happens when our project is divided into several sections. The model below shows the representation of sub function, sub application, and sub result.



Since our project is communication driven, it is important to assign individual function to employees or groups. When that happens, each function is considered as a part of the main function. We can group several parts of those functions into one function. We can also do the same process for application and application result. Below shows the representation of grouping parts of the application, parts of communication function, and parts of communication result in compressed form.



Grouping Communication Holder

Communication holder and external communication holder can also be grouped in the following form. Refer to the example section for more information. The shadow form can be used on a computer if necessary.



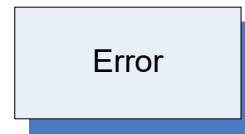
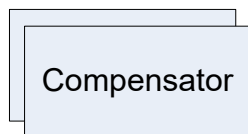
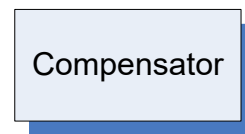
Grouping Entity and Communication Entity

Entities and communication entities can also be grouped in the following form. Refer to the example section for more information about the grouping entity.



Grouping Error and Compensator

Similarly to grouping entity above, error and compensator can also be grouped in the following form



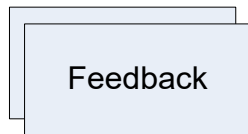
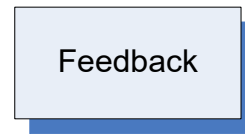
Grouping of Problem

As an entity itself, many problems can also be grouped in the following form



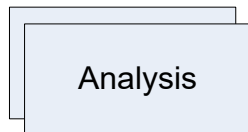
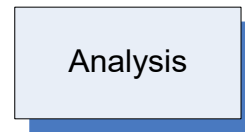
Grouping of Feedback

If needed, feedback can be grouped as shown by the model below. Assume that there is a need in a project to show all applicable feedbacks, and then the feedbacks received can be grouped on the form below. Refer to the example section for more information.



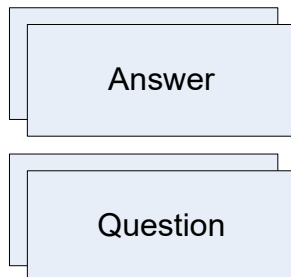
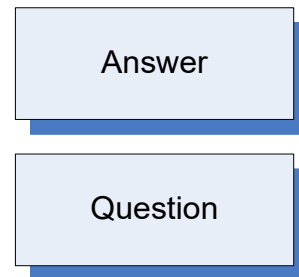
Grouping of Analysis

If there is a need in the project to show the list of analysis, then the analyses can be grouped in the form as shown below, see the entity usage section for more info.



Grouping of Questions and Answers

While it may not be necessary, however if there is a need to show a group of question and a group of answer, we can group questions and answers in the form below.

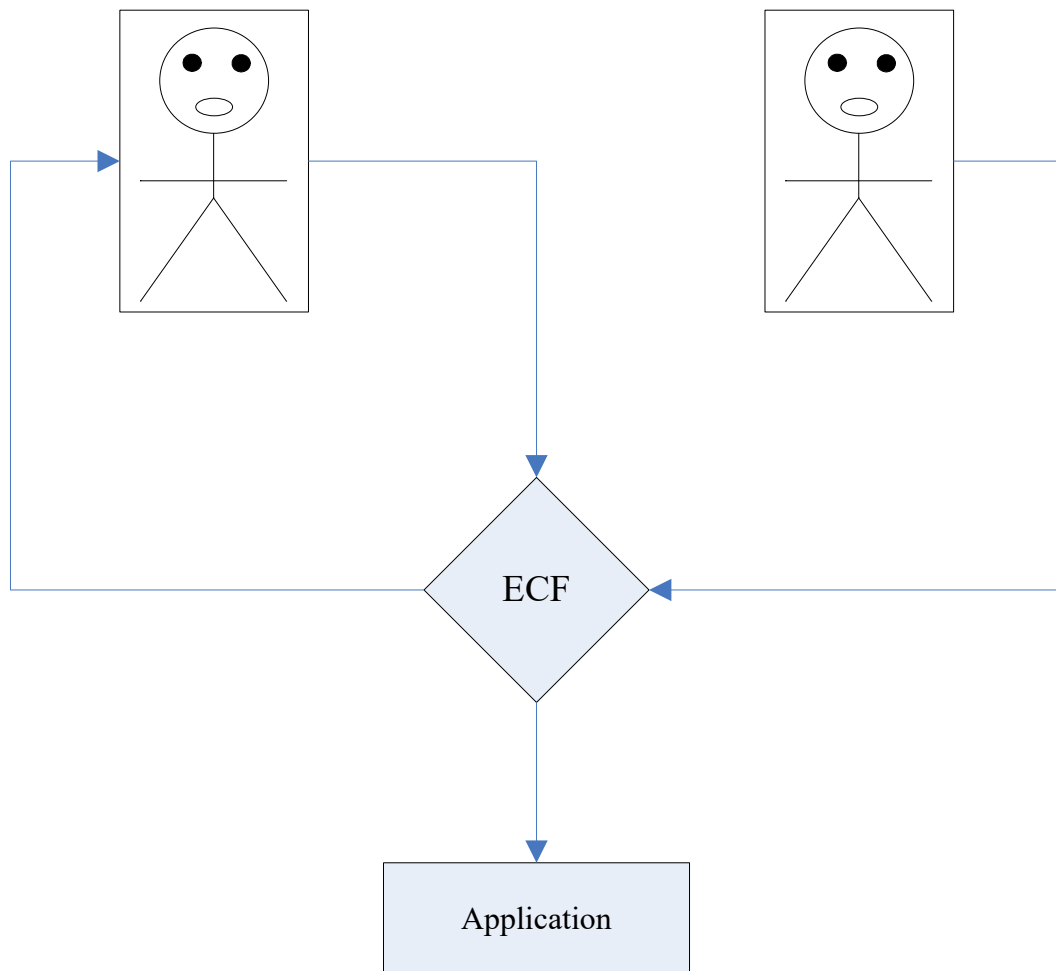


Some Entities Connection Examples

This section provides some examples of the usage of the entities. We can use this section to learn how to connect some of the entities in an actual project.

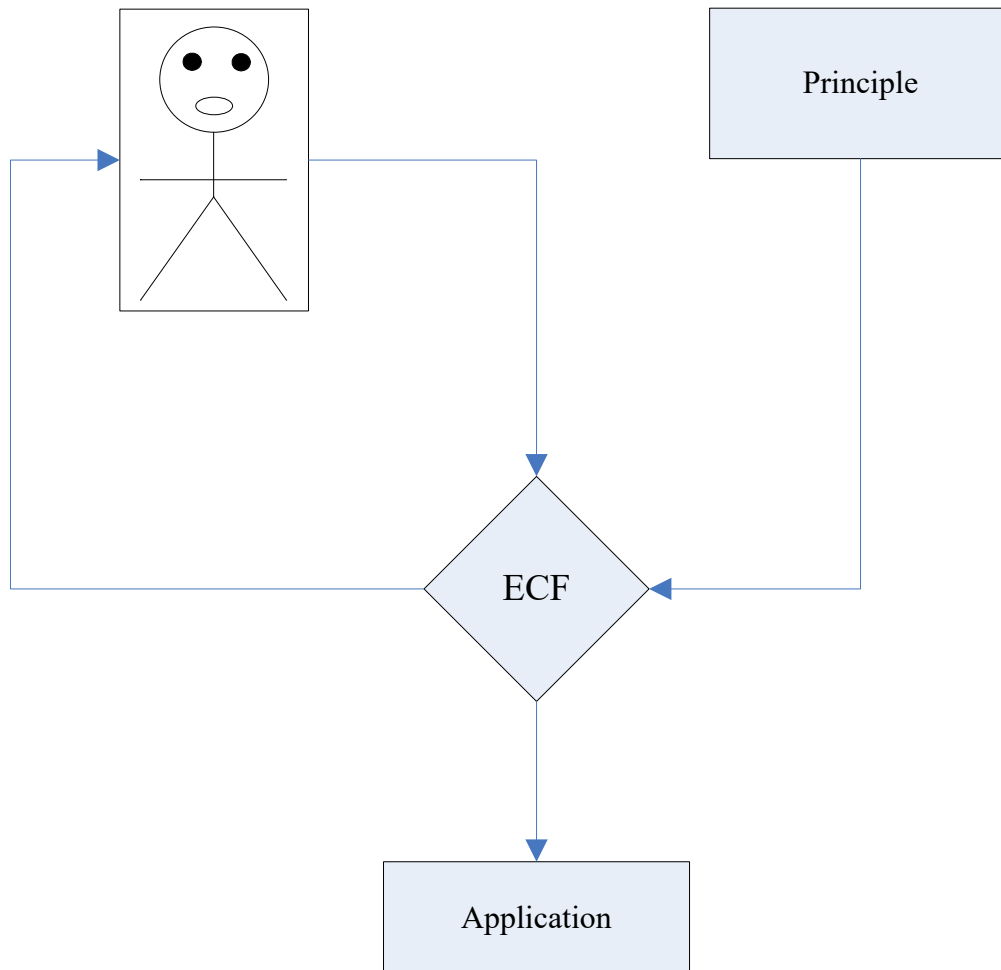
Example Number 1

Below we show the usage of the error correction function. In this case, the person to the left can be considered as the children while the one in the right as the parent.



Example Number 2

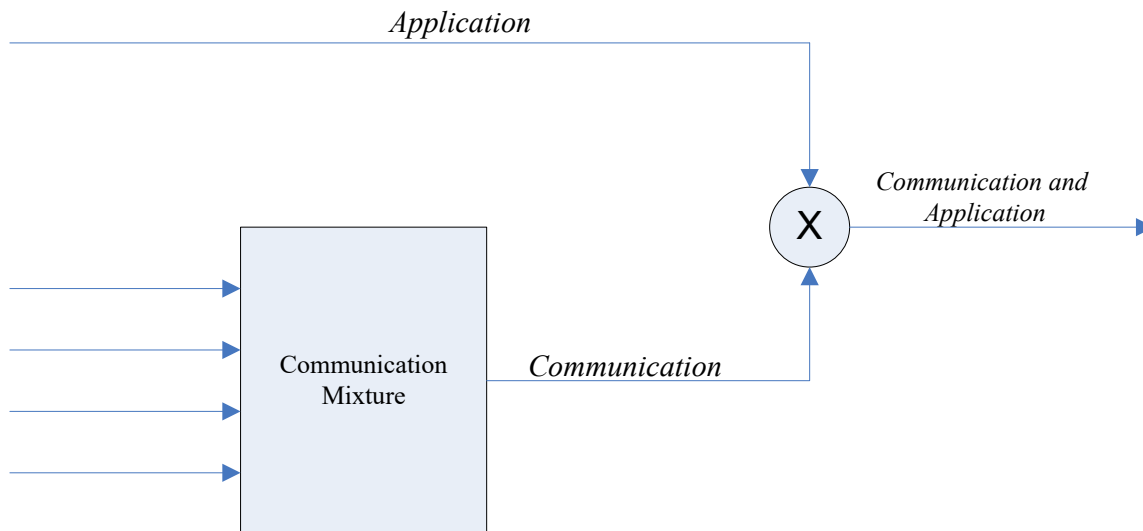
Since the principle is considered to be the parent itself, then the principle can replace the parent on the model above. The model below is the same as the one in the example above, except the parent is simply replaced by the principle.



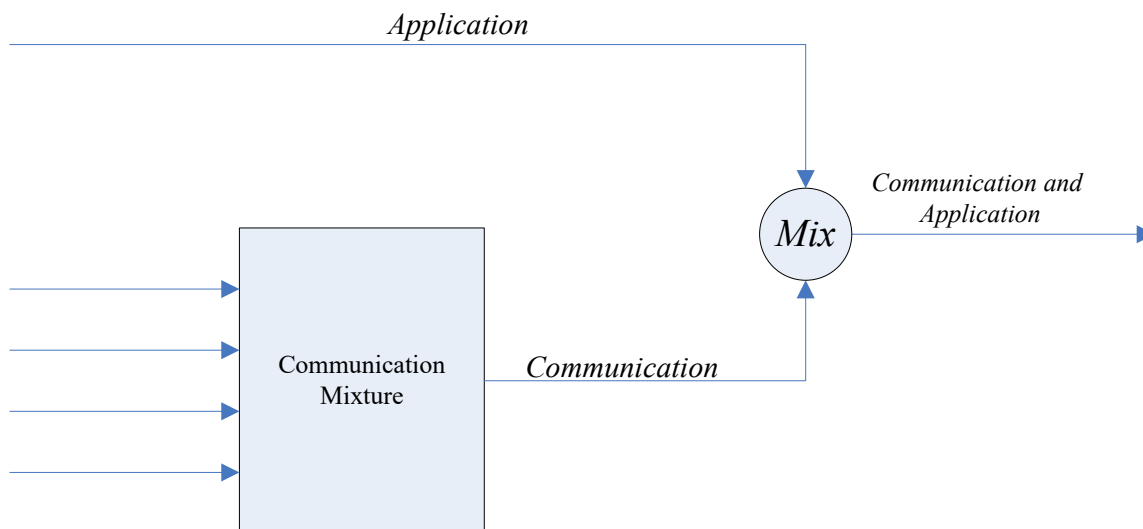
Example Number 3

The model below shows the usage of the communication mixture entity. As we know, the communication mixture can take several communications as input. The output of the communication mixture then feeds the communication application mixture as shown to the right. The way to look at it, communications from many people can be connected to the communication mixture entity. In this case, the output of the communication mixture entity

contains communications of those people. On the model below, communication from four people are connected directly to the communication mixture entity.

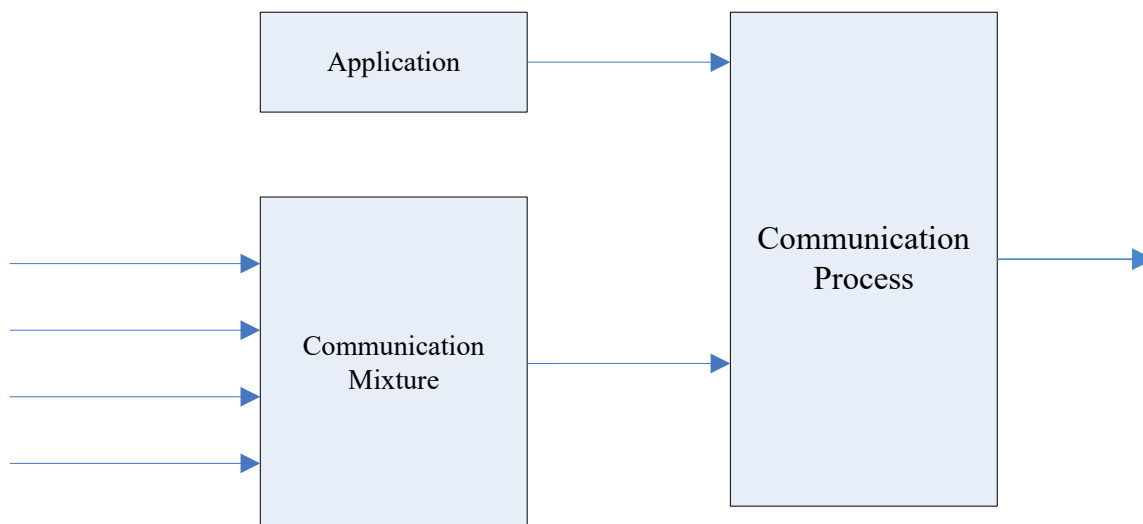


The model below is the same as the one above, except the **X** inside the circle is replaced by Mix. Below we use the *Mix* to denote the mixture of the communication and the application. It does not matter; we can simply use the whole word or the phrase.



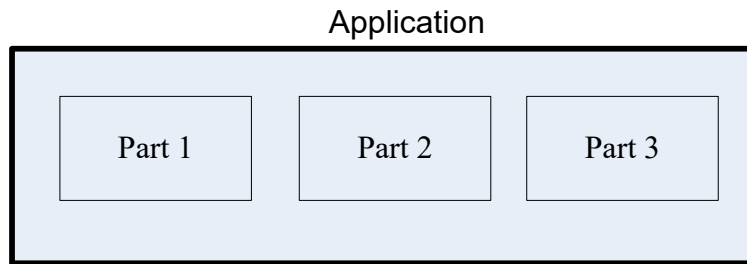
Example Number 5

We do have option to use the communication process entity or the communication mixture with the communication application mixture. It does not matter which one we use, we can still use the communication mixture entity to group our communications. The communication mixture entity can also feed the communication process entity. We can also connect people or communication directly to the communication process entity. The way to look at it, the communication process entity includes inputs for both communication and application. It is always good to group all of our communications as a single entity to feed the communication process entity. The model below shows the usage of the communication mixture as a single entity to feed the communication process entity. The communication process entity is the same as **Mix** inside the circle or the **X** inside the circle. The model assumes the separation of the application from communication. It is also good to understand that other communication mixture can also feed the communication mixture entity.

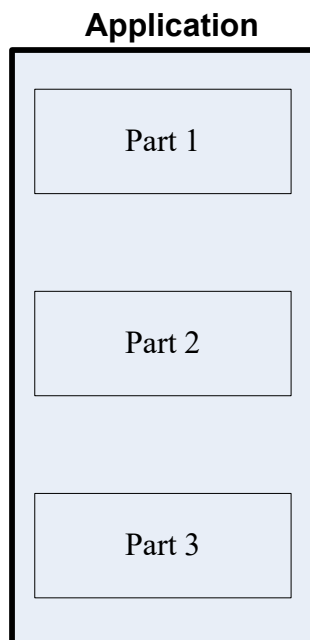
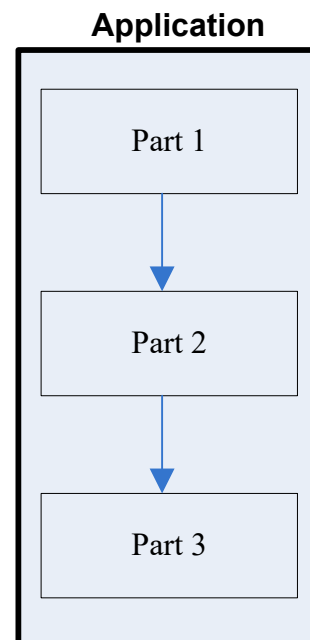


Example Number 6

If the application is divided into several sections, the application entity can be used to show that. Below the application is made of 3 sections, we use parts of application to show that. In this case, each section is considered to be a part of the application. We can also use the grouping entity and the empty container to group part of the application.



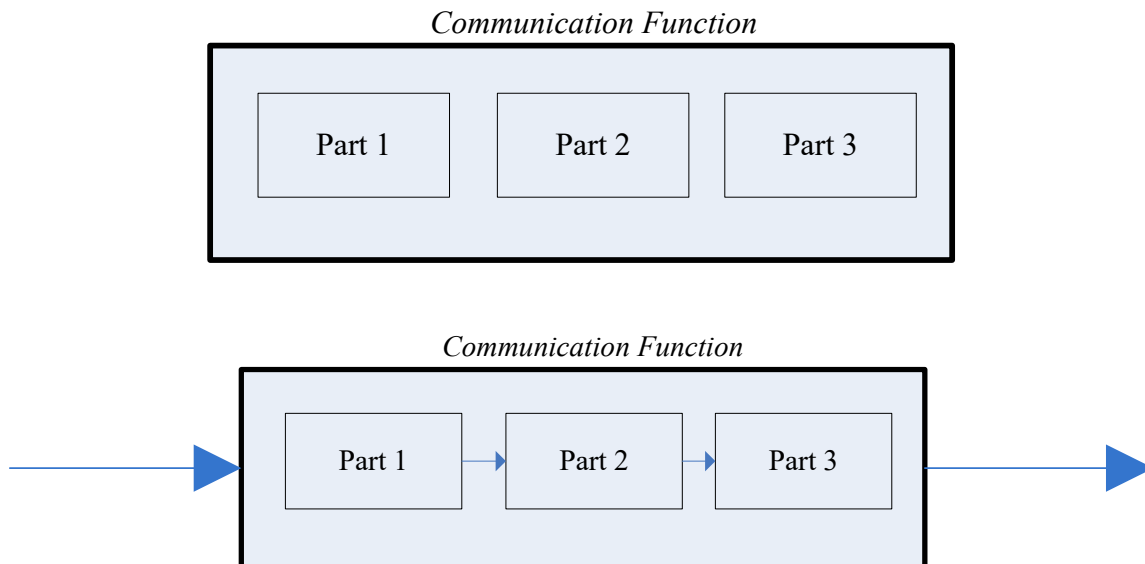
Instead of using the grouping entity as shown above to show the grouping of several parts of the application, the grouping of several parts of the application can also be presented in the following form as shown below. This case depends on the type of paper or drawing board that is being used to model the application. Assume that we are using a portrait layout of paper, where we don't have a lot of width to work with, then we can use the portrait form of the application to group the parts of our application. If we want to, we can use arrow to show how the parts are connected together. Below, the model to the left is the same as the one to the right, except in the one to the right, we use arrows to show the connection of the parts of the application.



Example Number 7

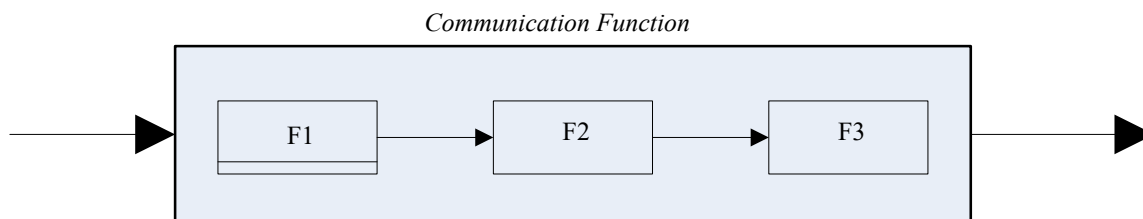
Compared to the grouping of the application shown above, for applications that are divided into several parts, those parts can be grouped as shown below. We mean an application where the communication function is divided into several parts or sections. Both the first model and the second model can also be used to show the grouping of parts of the communication function. The way to look at it, the bigger entity represents the overall communication function, while the

small entities represent parts of the communication function. We can also say that the bigger entity represents the overall parts of the communication function. Below, the second model is the same as the first one, except in the second model, we use arrows to show how those functions connect together.

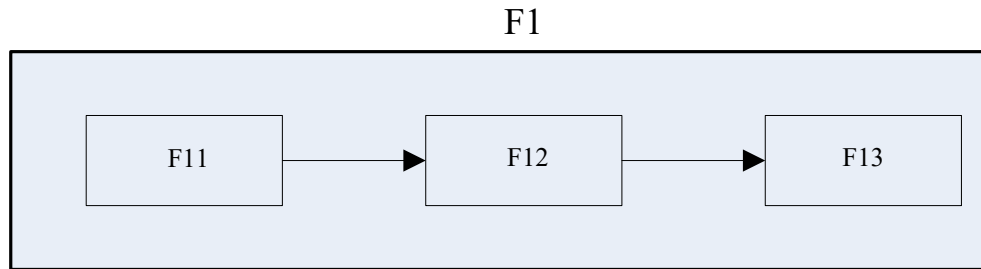


Example Number 8

If the communication function is made of several sections, sub functions can be used to show parts of the function of the communication as well. Here is an example; in this case F1 is a sub function. This example is similar to the one above, except one of the functions is considered to be a sub function. A sub function is a function that contains other functions inside. If the communication function is divided into several parts, some parts of the communication function can also be considered as sub functions.

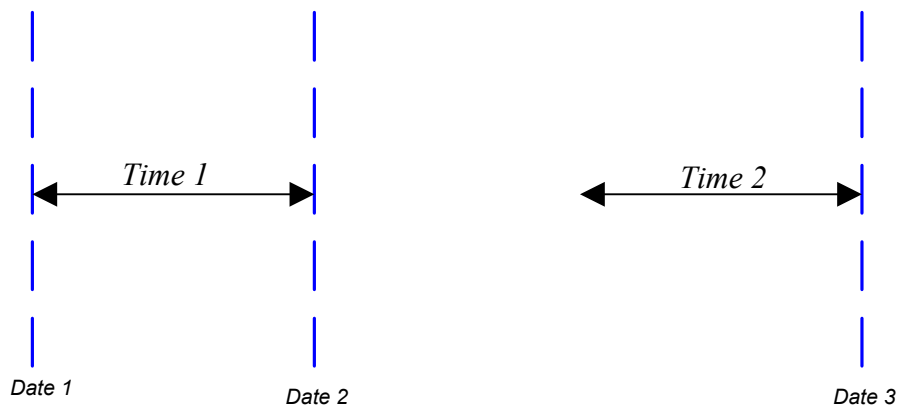


The model below shows the expansion of the sub function **F1**. As we have said it, a sub function is a function that contains other functions inside. As shown by the model below, we can see that the sub function **F1** above expanded to functions F11, F12, and F13. All those functions are considered to be parts of the main communication function.

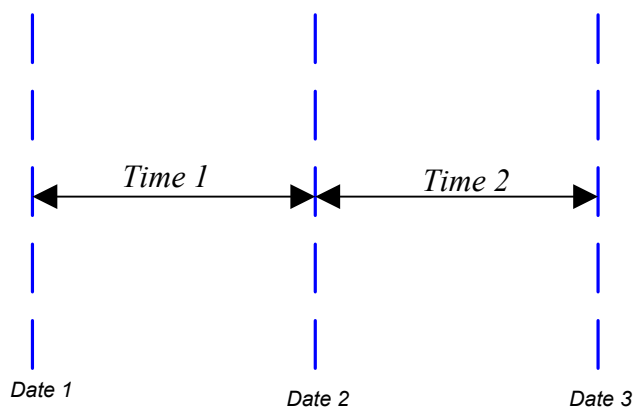


Example Number 9

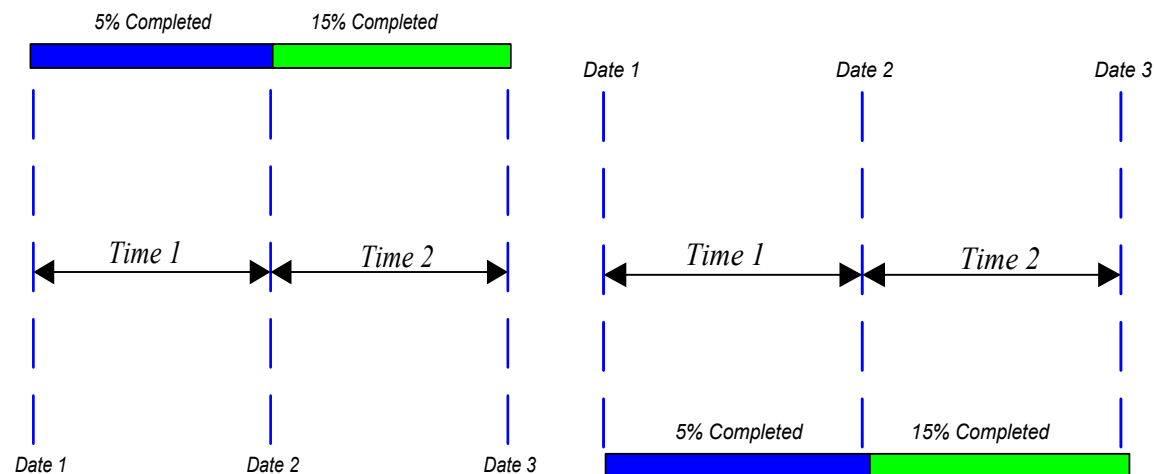
Time line can be used in the following form to show function execution related to time or to show specific process related to time. For instance, we can use time line to show the execution of our communication function or the time of a communication. The first model below shows the first time line while the second one to the right shows the successive time line.



The model below is the same as the one above, except that we connect both time lines together to show an overall single time line. All that we do connect the first time line with the successive time line.

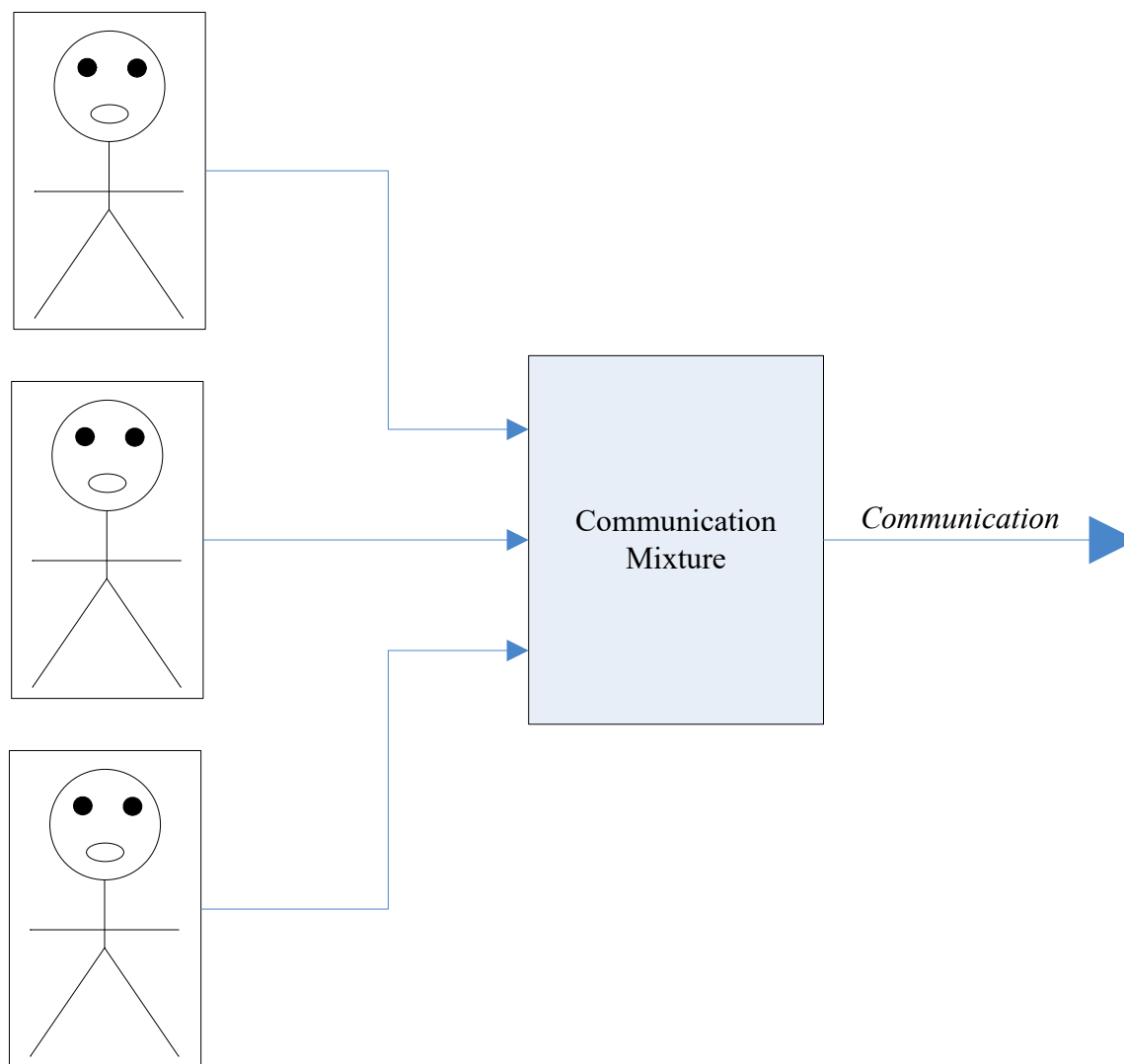


We can also use progress bar with time line to show the progress of our application. In this case, we can use the progress bar with specific time line to show the progress of the application at a particular time. The model below uses both progress bar and time line to show the application progress at a given time. In this case we can see at *time 2* we are 15% completed. The model to the left is the same as the one to the right, except in the one to the right, the progress of the application is shown at the bottom.

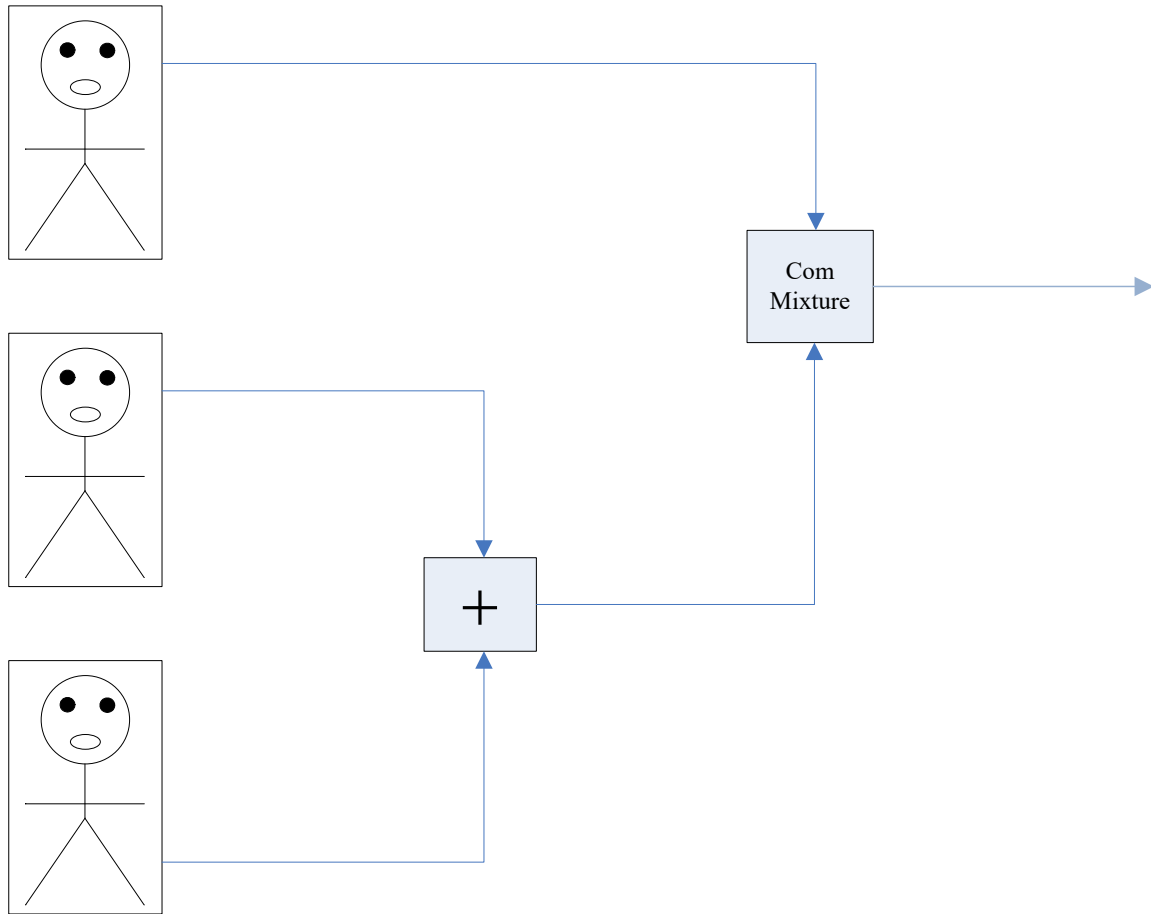


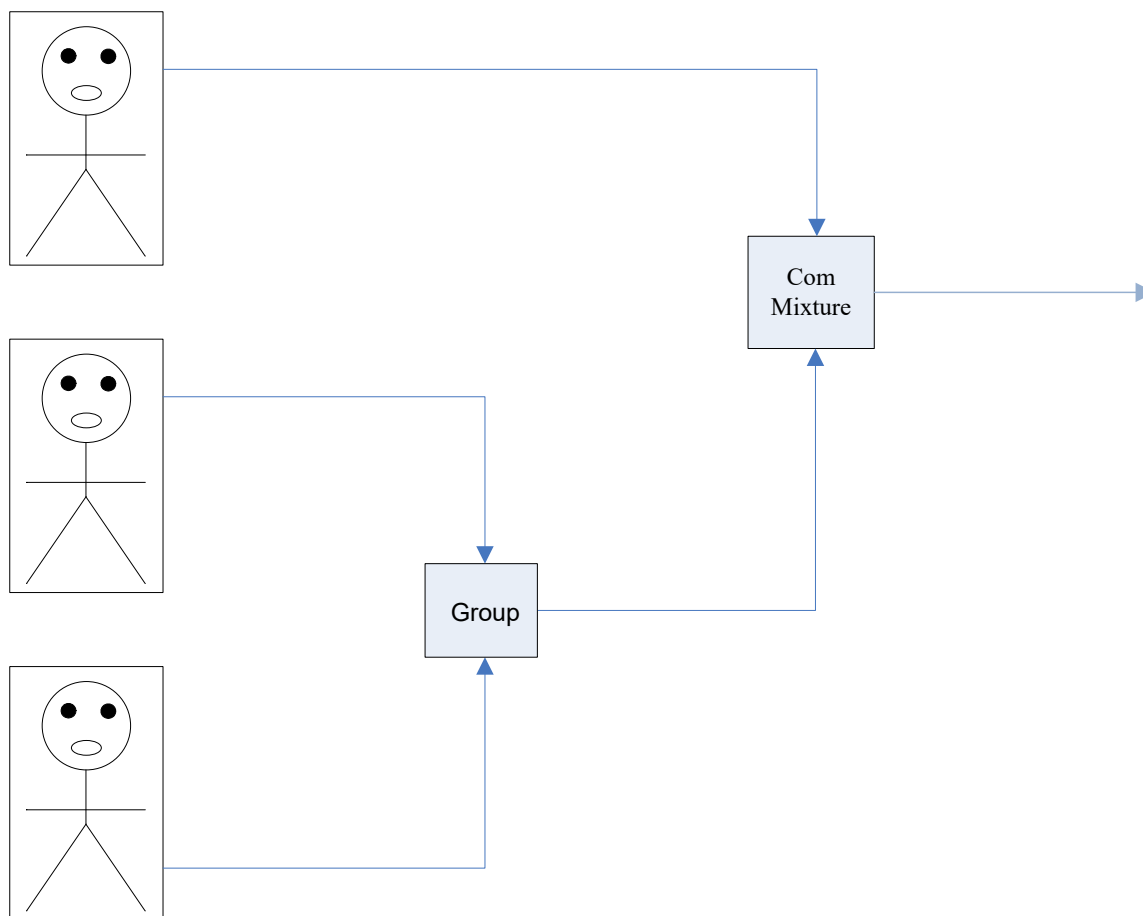
Example Number 10

We know that the communication mixture entity takes communication as input. In this case, we can connect people directly to the communication mixture entity to show how their communications feed that entity. In the model below, we connect three people directly to the communication mixture entity. The output of that entity is the total communications of those people.



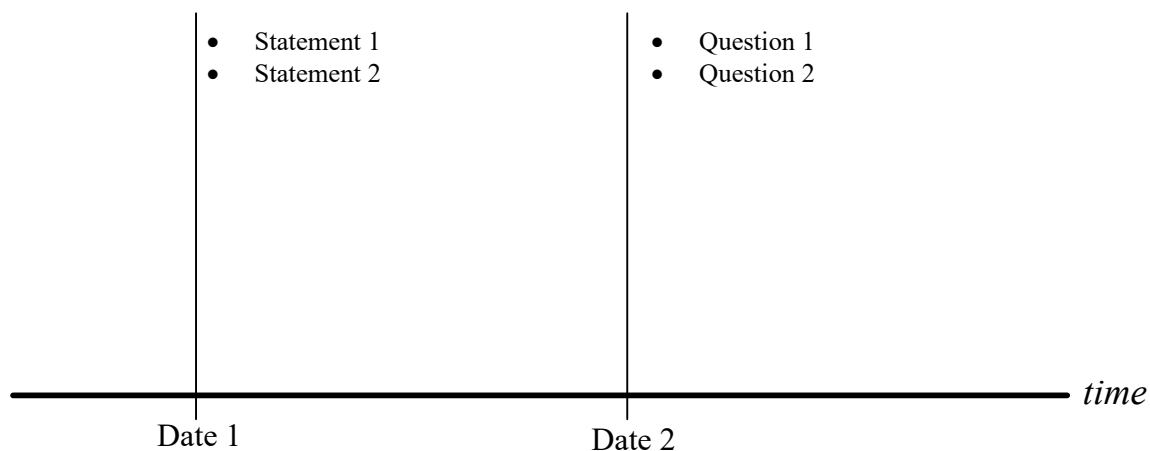
The model below is similar to the one above, except the grouping entity is used to show a group of two people. We know that we can use the grouping entity to group people; we can also use it to group communication mixtures if we want to, although it is not necessary. The way to look at it, while the grouping entity is used to show a group of two people; however the communications of those people feed the communication mixture entity. Again, the output of the communication mixture entity includes the total communications of those three people. The second model below is the same as the first one, except *group* is being replaced by the plus sign instead. It does not matter; we can always change the text in an entity.





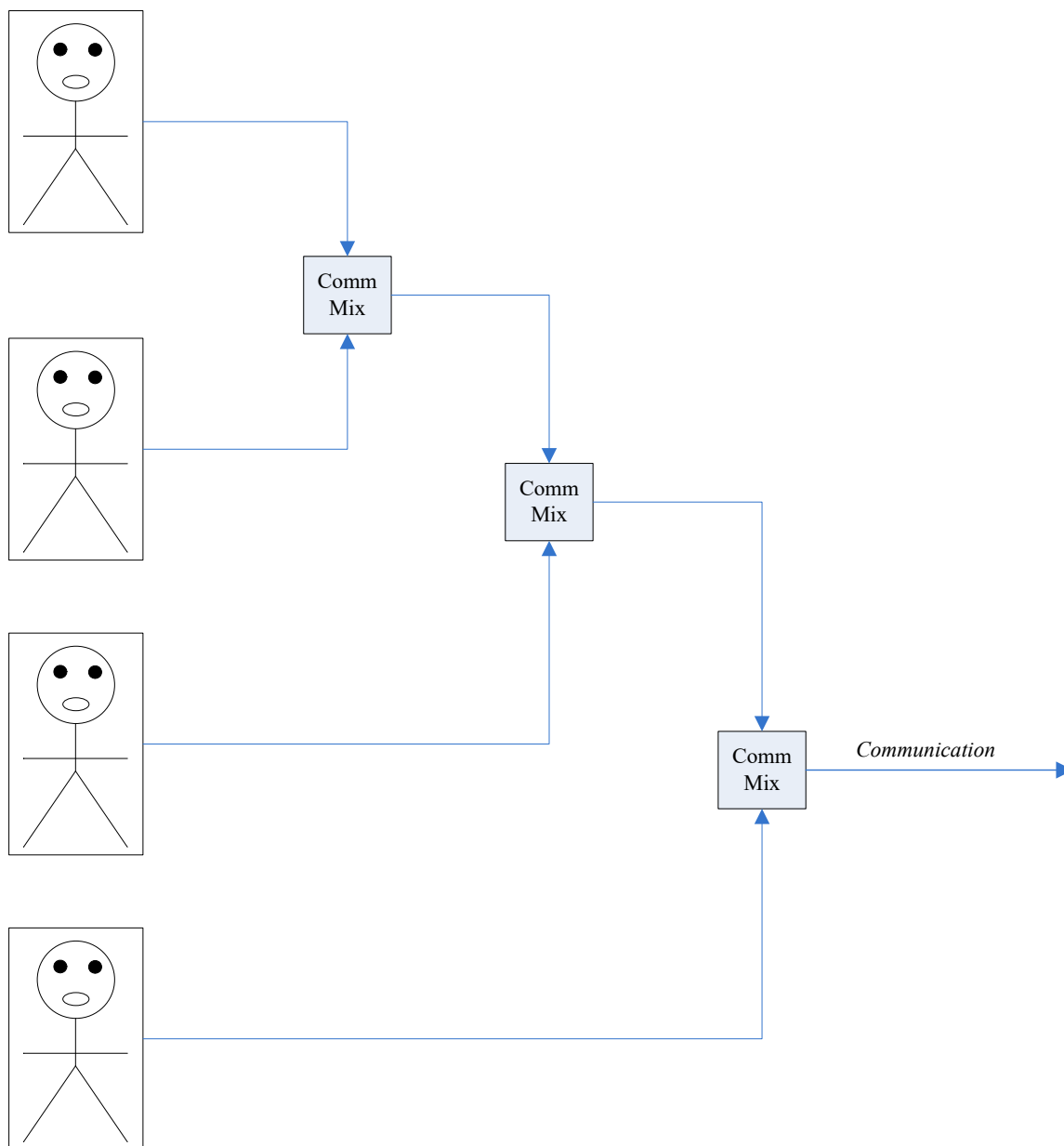
Example Number 11

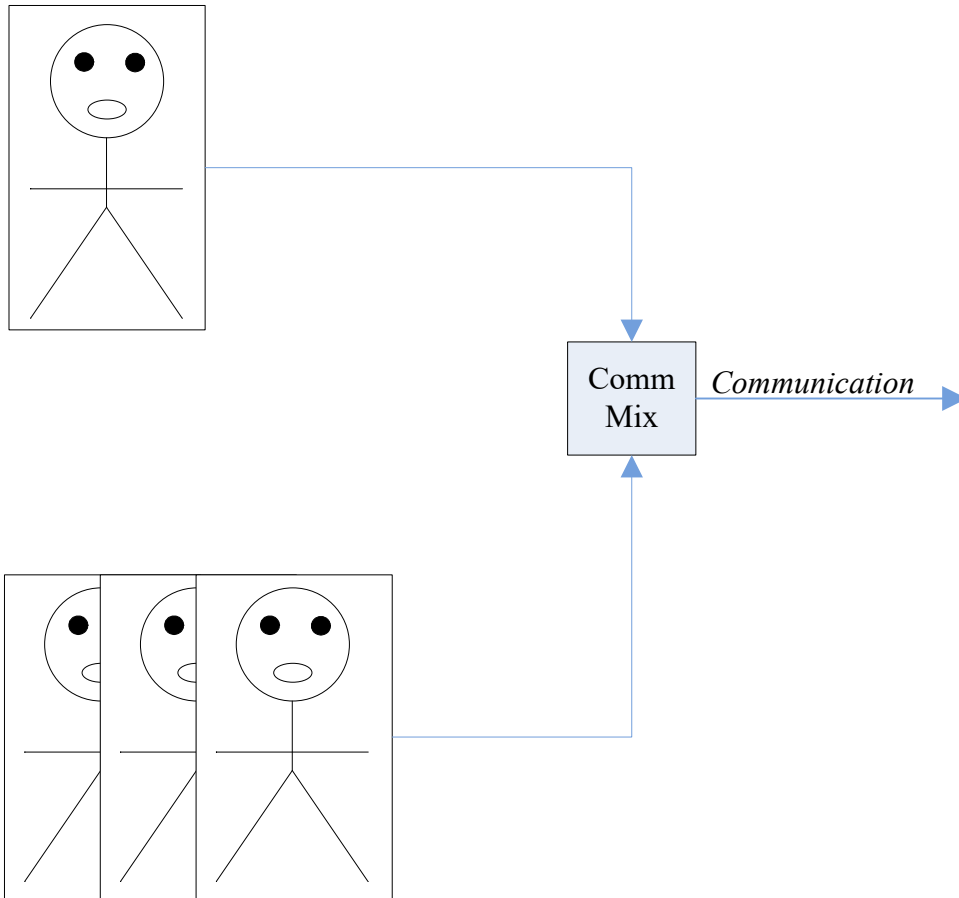
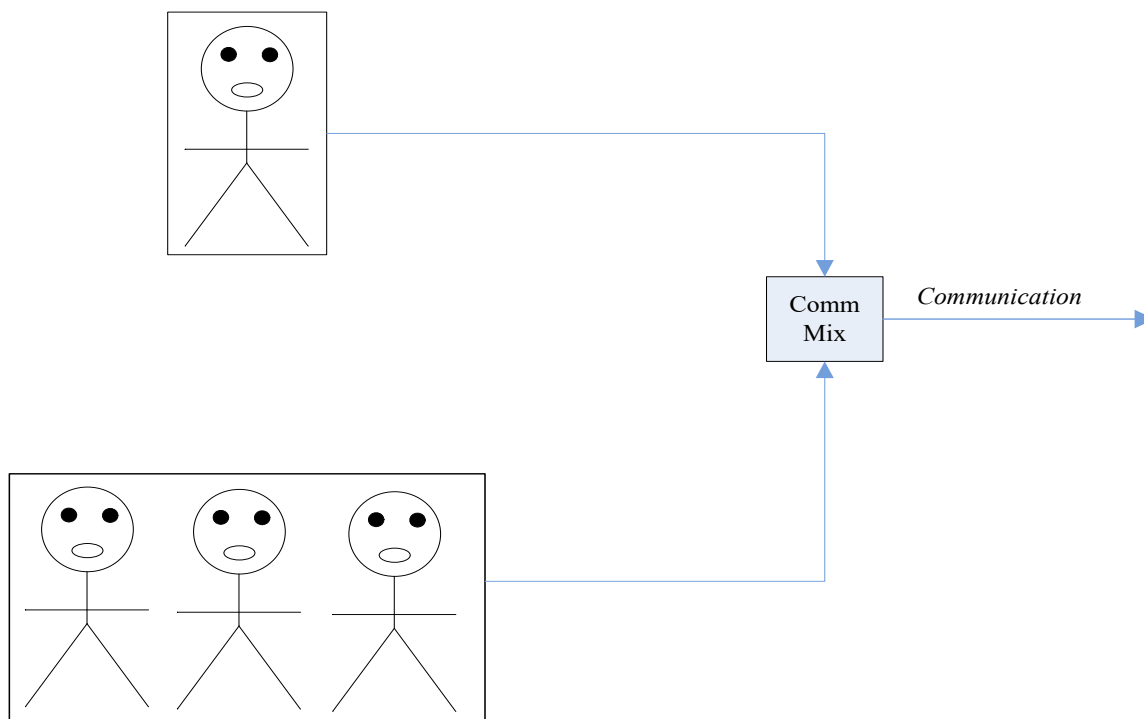
We can use timeline in conjunction with dateline to show specific statement or question at a given date or time. Below is an example of using the time chart with the dateline to record statements and questions. From the model below, Date 1 is the recorded date for statement 1 and 2, while Date 2 is the recorded date for question 1 and 2.



Example Number 12

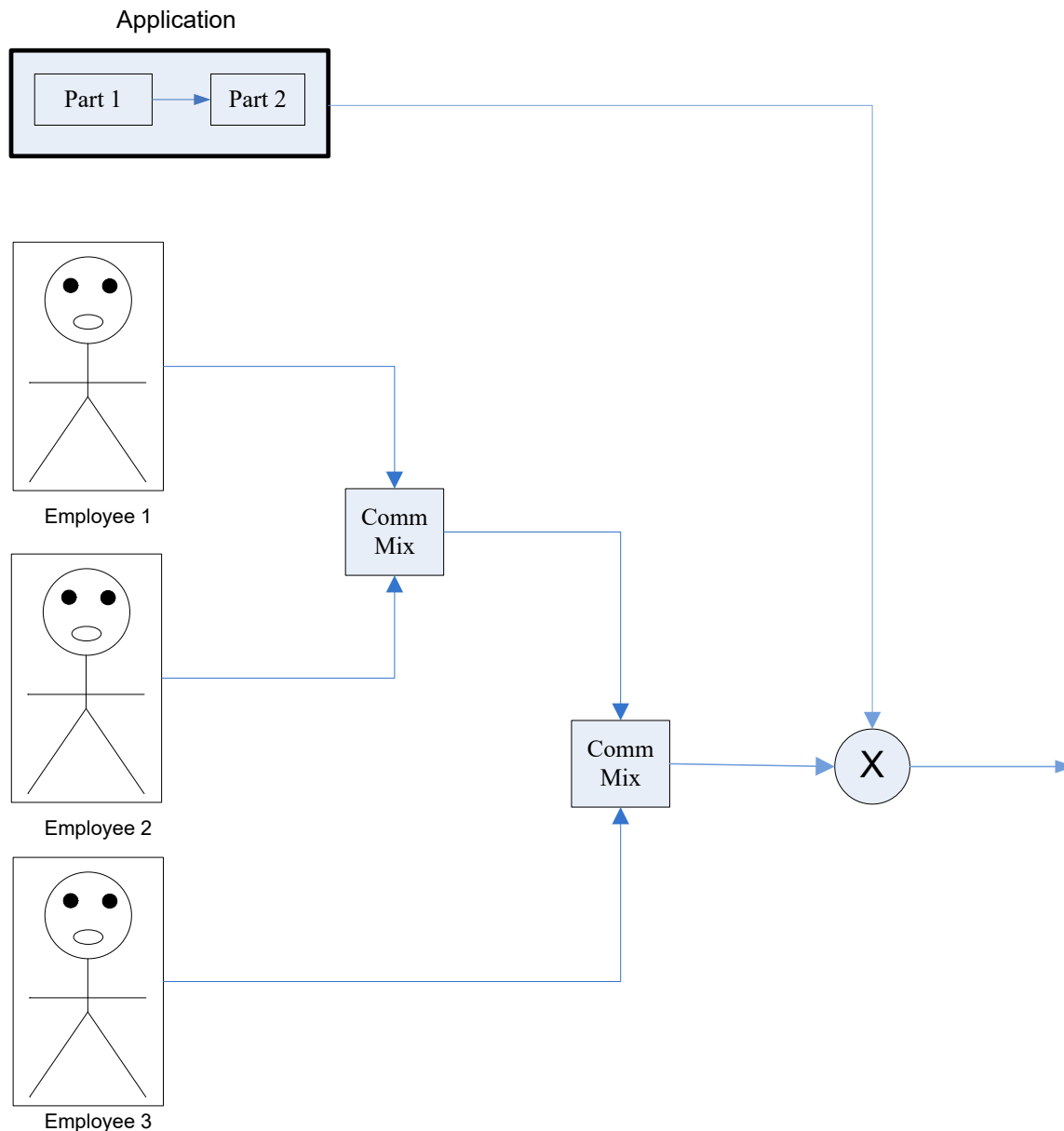
The model below shows the starting communication of two people that extends to four people. As usual, the output of the communication mixture entity includes the total communications of the people that feed that communication mixture. The last output of the communication mixture entity includes the communications of all people in the project or that communication. The model below shows a total communication of four people. To reduce space, we can group some of those people together. The second model below simply shows the grouping of three of those people. Keep in mind that the second model is the same as the first one, except in the second model we group some of those people together. The output of the communication mixture is still the same. The third model below is the same, except we use the linear form of grouping.





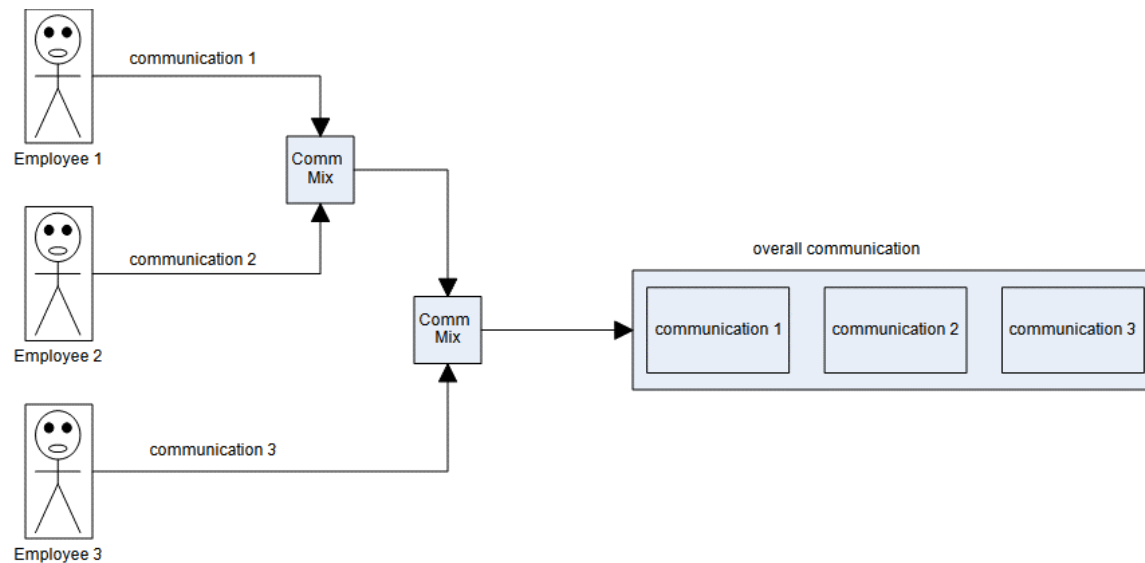
Example Number 13

Assume that our application is divided into two parts, we can use part of the application entity to show that. The model below shows that our project is made up three employees, where the application is separated into two parts.



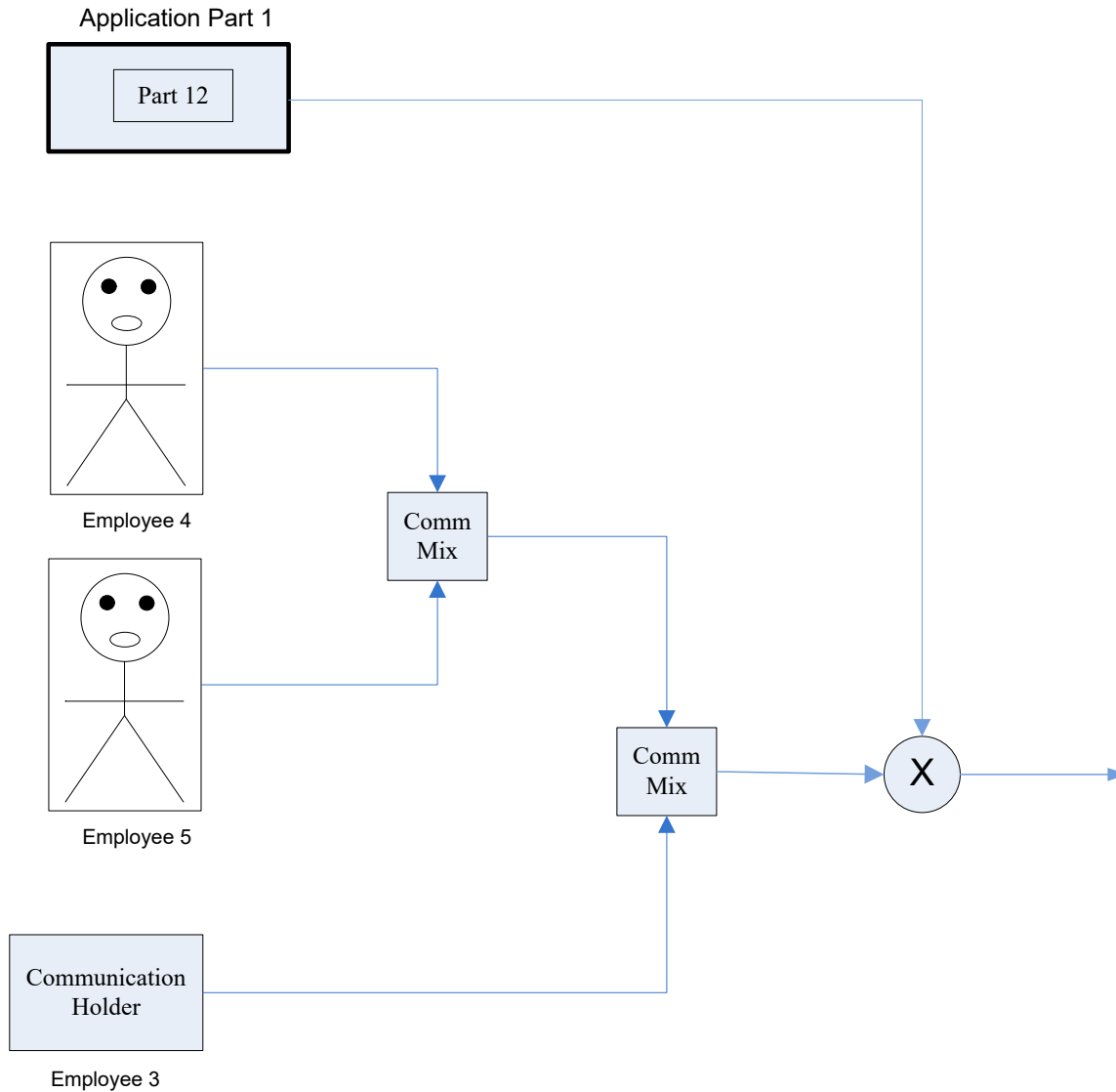
We can also represent the communication for each employee in this form by identified the communication of the employee next to the employee. We can associate the communication of

the employee with a number to denote that employee communication. Below is an example, communication 1 is the communication for the employee 1, communication 2 is the communication for employee 2, and communication 3 is the communication for employee 3. The overall communication of all the employees is shown to the right.



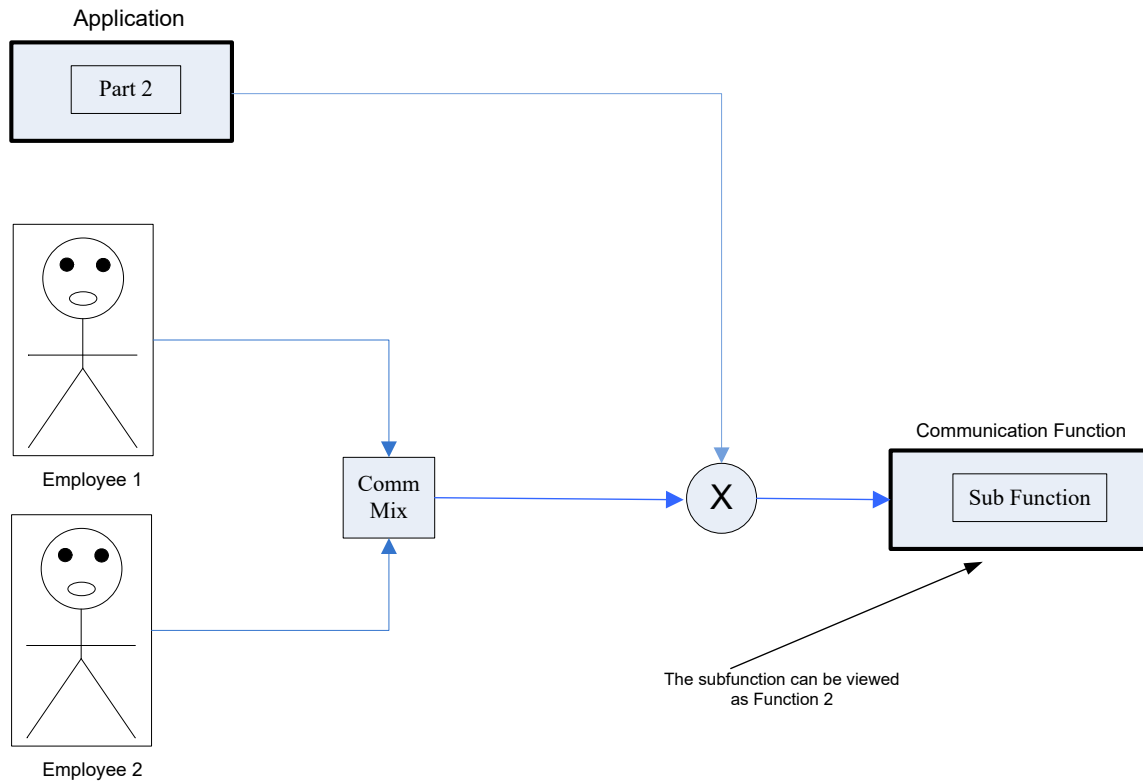
Example Number 14

This example is similar to the one above; except that employee number 3 is being replaced by a communication holder and part 1 of the application also include other parts. In this case, it looks like part 1 of the application is viewed as a sub application. The way to look at it, while employee three is being worked in a project that has different parts; but employee three can also be worked in another part of the project. Since employee three cannot be duplicated physically, in this case when the whole project is being viewed in the same sheet or on the same screen, it makes sense for employee three to be replaced by his/her respective communication holder. For instance if we model our project on a piece of paper, where employee three has two functions in different section of the project, when both sections are being viewed in a single sheet or on the same screen, in one section employee three is replaced by his/her respective communication holder as shown below.



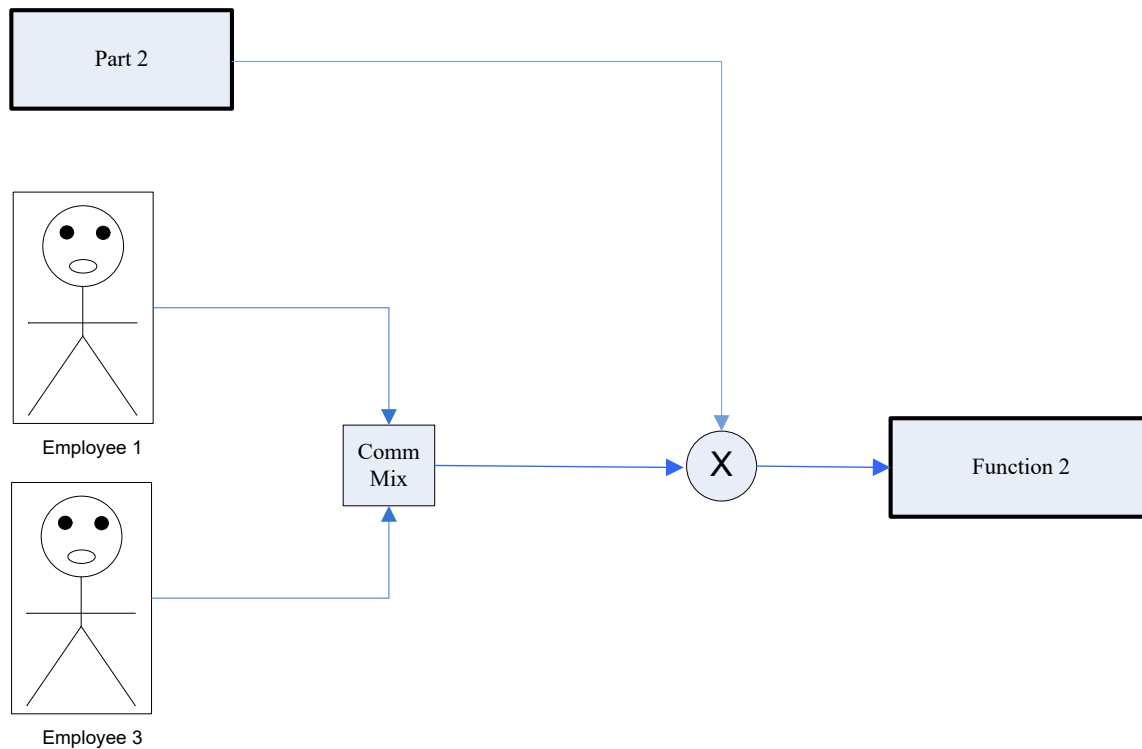
Example Number 15

This example is similar to the one above. In this example, the communication function includes a sub function, where the sub function can be viewed as function 2. We can use sub function to group part of the communication function that can be outside of the main function. We can also use them to show functions that are also parts of the main function.

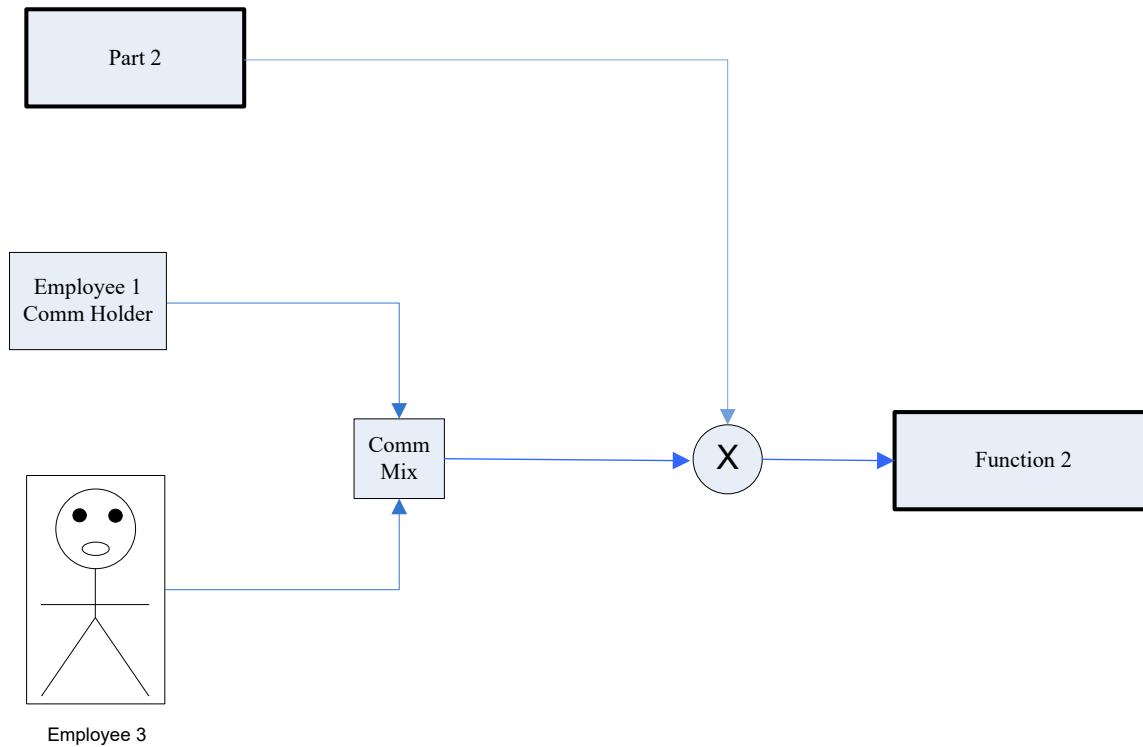


From the model above, the sub function is considered to be function 2. Now, we can expand the sub function to show more information about this section of the project. By doing so, we can

also see the part of the application is also expanded with the sub function. In this example, part 2 of the application is also linked to function 2, where function 2 is considered the main communication function for that part of application. Our model below assumes that our project is made of several sections. In this case, each section can lead or managed by different people. For this reason, it makes sense to have sub functions, part of functions, sub results, parts of result, sub applications, and parts of application.

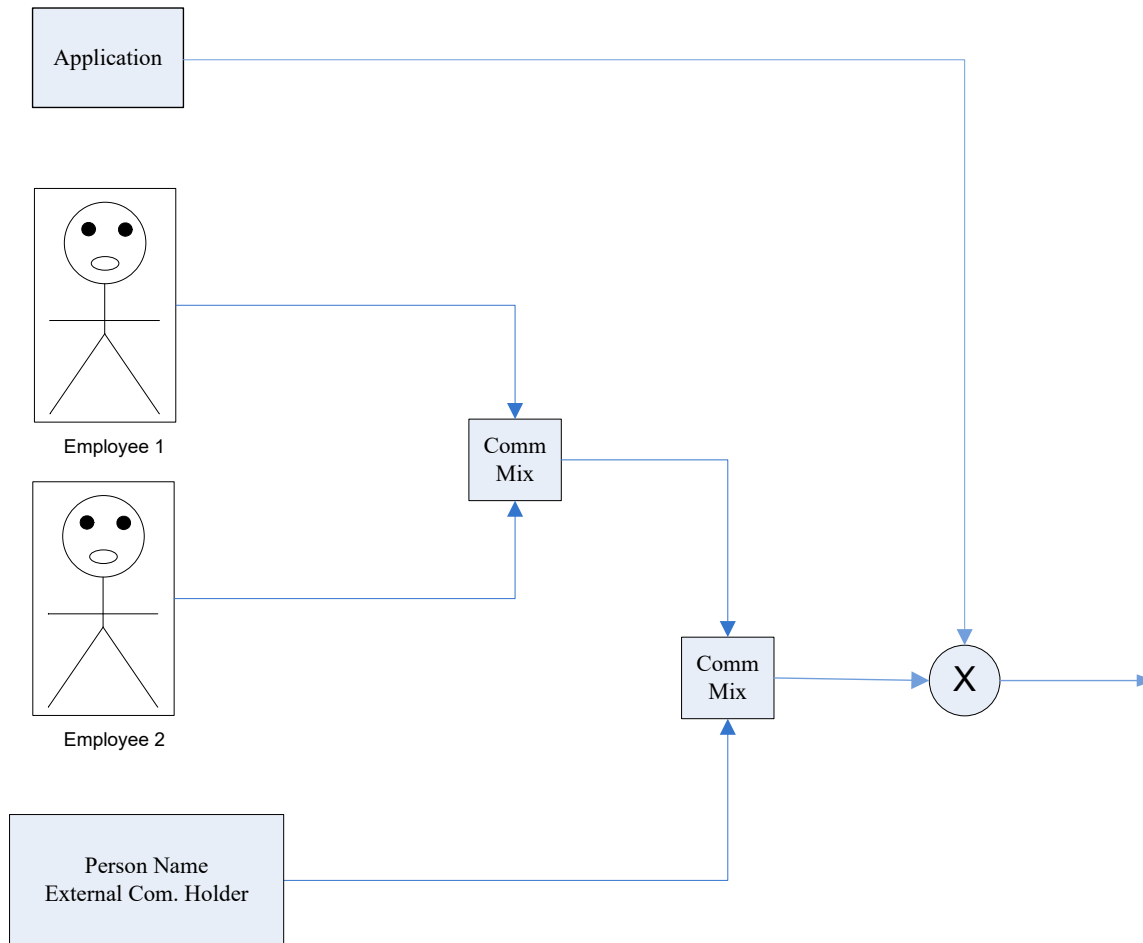


The model below is also another way of looking at the model above. If we assume that employee 1 can be working on a different part of the project, but managed by another person. In this case, when viewing both models together, it makes sense for employee 1 to be replaced by his/her respective communication holder. Assume that employee one is working and managed by one person, but helping out another project that is managed by another person. In this case, when both projects are linked together and being viewed in the same screen or paper, in one of them, mostly the second one, employee 1 will definitely be replaced by his/her communication holder. The model below simply shows that. While we say help out in another project, we mean help out in another part of the project. It is always good to view the overall project as a single project that has different parts, but managed by different people.

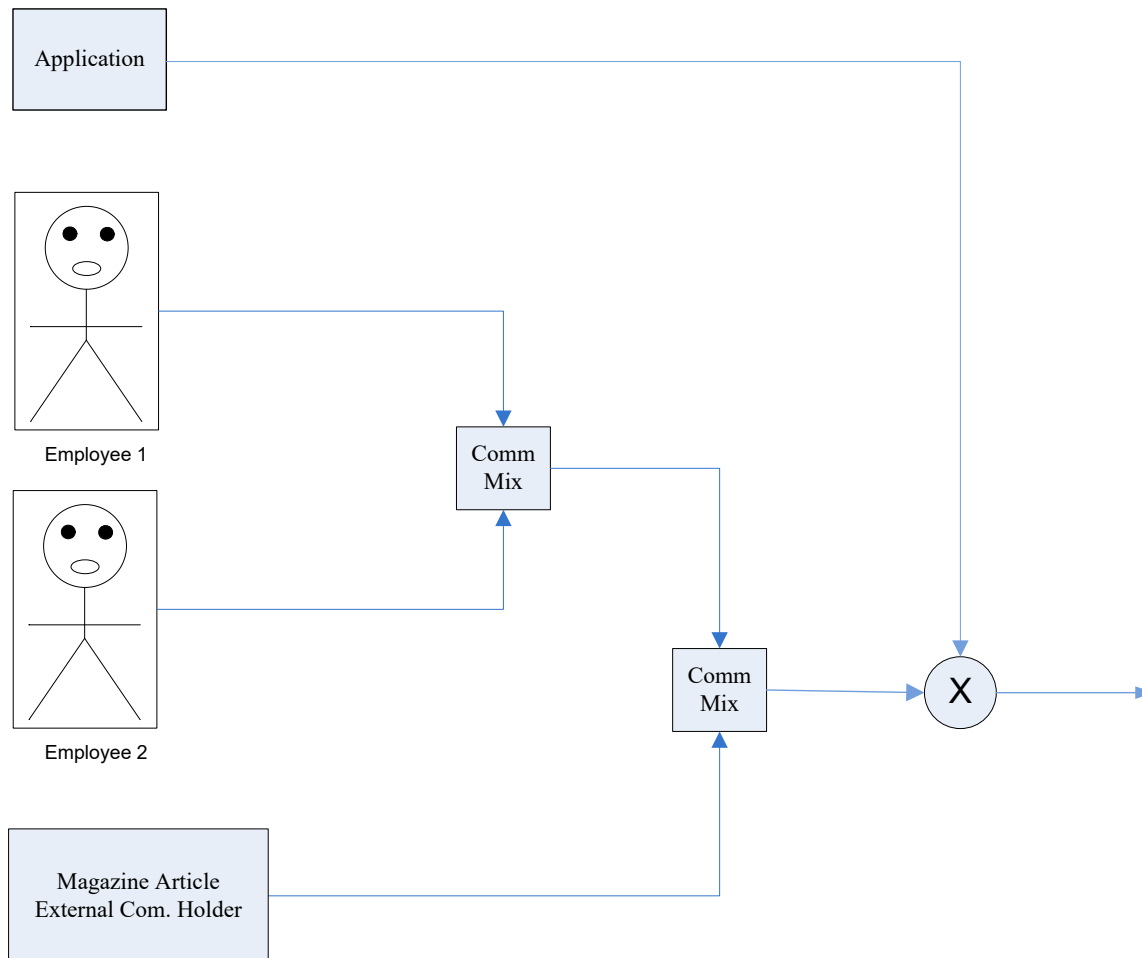


Example Number 16

In a project, it may be possible for us to use communication of people that are not physically in the project. The external communication holder enables us to do so. It is very important for us to take communication into consideration in our project. The external communication holder can hold communications from external sources. For instance people, books, magazine articles, newsletters, interviews etc. The model below shows the usage of the external communication holder. In this case, it shows the communication of a person that is not physically in the project. While that person is not in the project, however its communication contributes to the project.

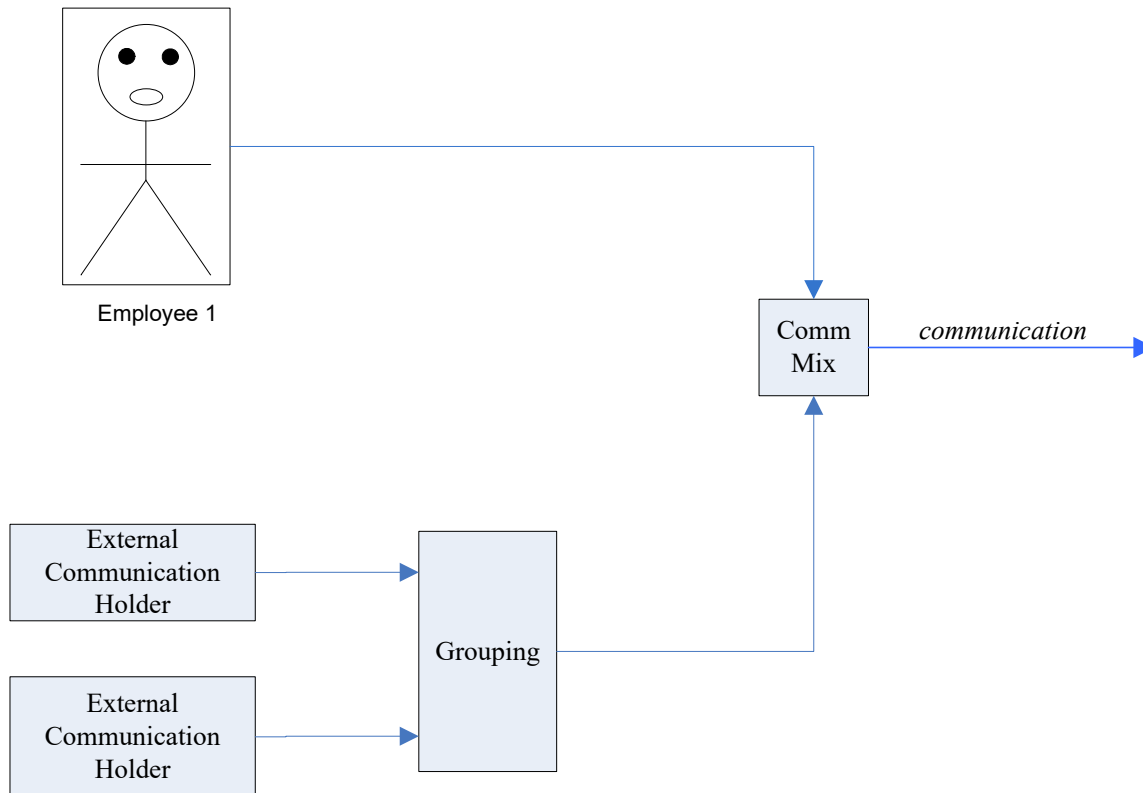


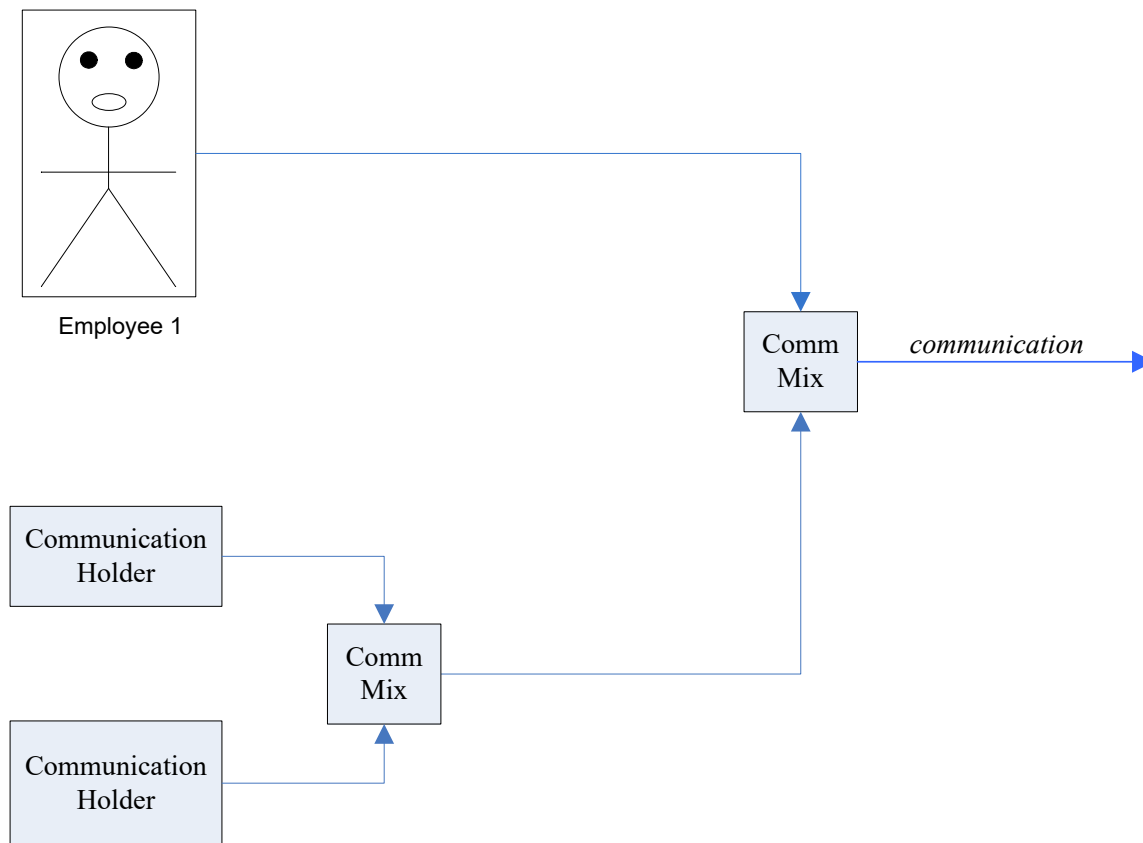
The model below is similar to the one above, except the external communication holder holds communication from a magazine article. Since our project can contain communication from different sources, in this case we use that magazine article in our project. That magazine article can contain information that enables us to execute the overall function of the project, which is the communication function. That magazine article can also affect the result of the project. It is very important to take external communications into consideration in term of analysis.



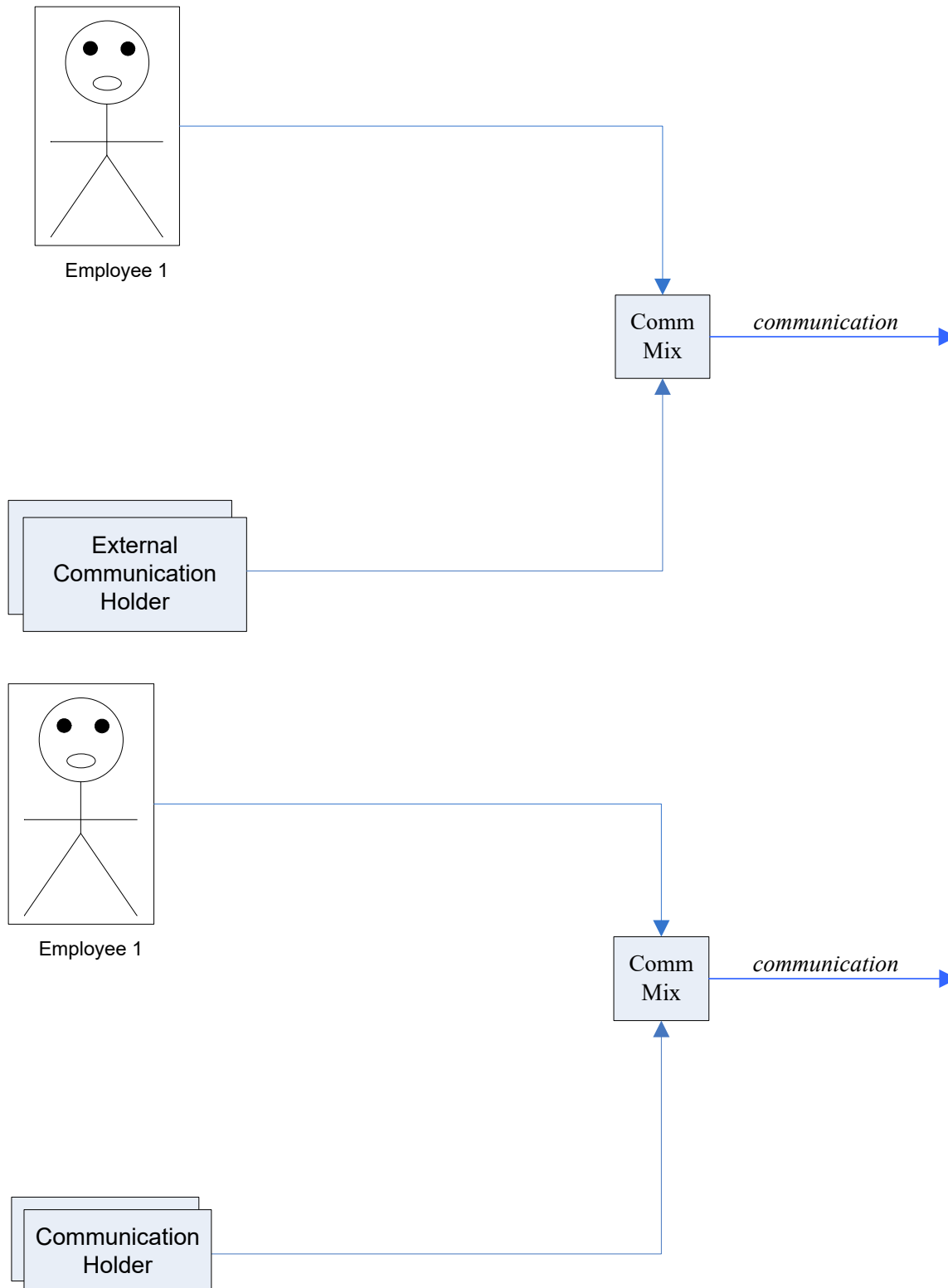
Since external communication holders are considered to be entities, they can also be grouped by using the grouping entity. From the model below, we use the grouping entity to group two external communication holders. Since the project we are working on depends on what we need to do, there is no limit on how many external communication holders that can be used. Assuming that we are working in a project where one piece of information comes from a book and one from a magazine article, we can group both of them to show that. We can also group them together as long as they can be combined to enable us to do what we do. From the model below, we group two external communication holders. In this case, we assume that both of them

are related to enable us to do what we do. The second model below is the same as the first one, except we use the communication mixture for the two external communication holders rather than using the grouping entity. Since external communication holders are considered to be communication themselves, they can be mixed together using the communication mixture entity.





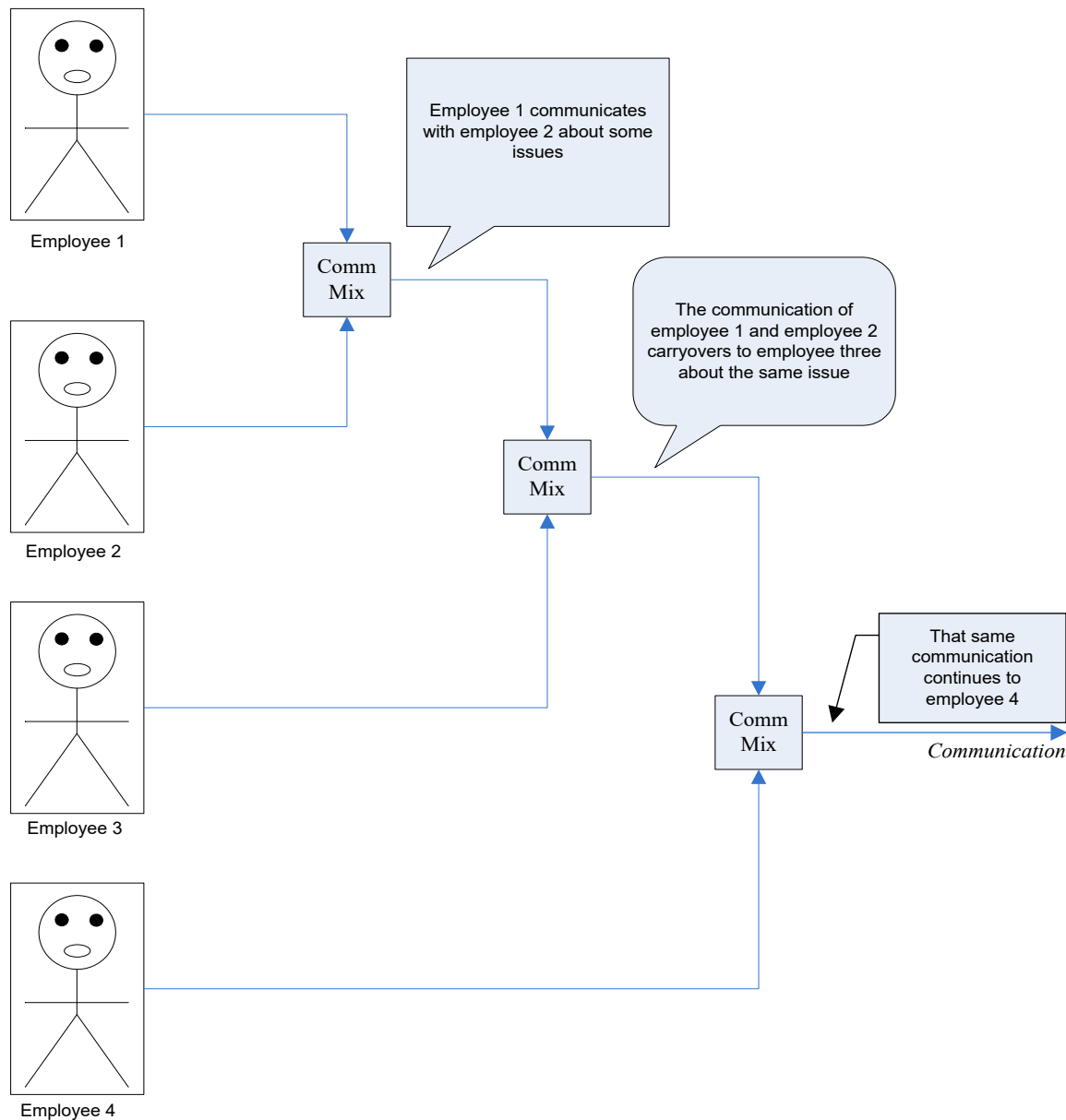
Since communication holders are considered to be entities themselves, they can be grouped in the same form as we group entities. The models below show another way of grouping external communication holders and communication holders. We can use this form of grouping in order to preserve space.



Example Number 17

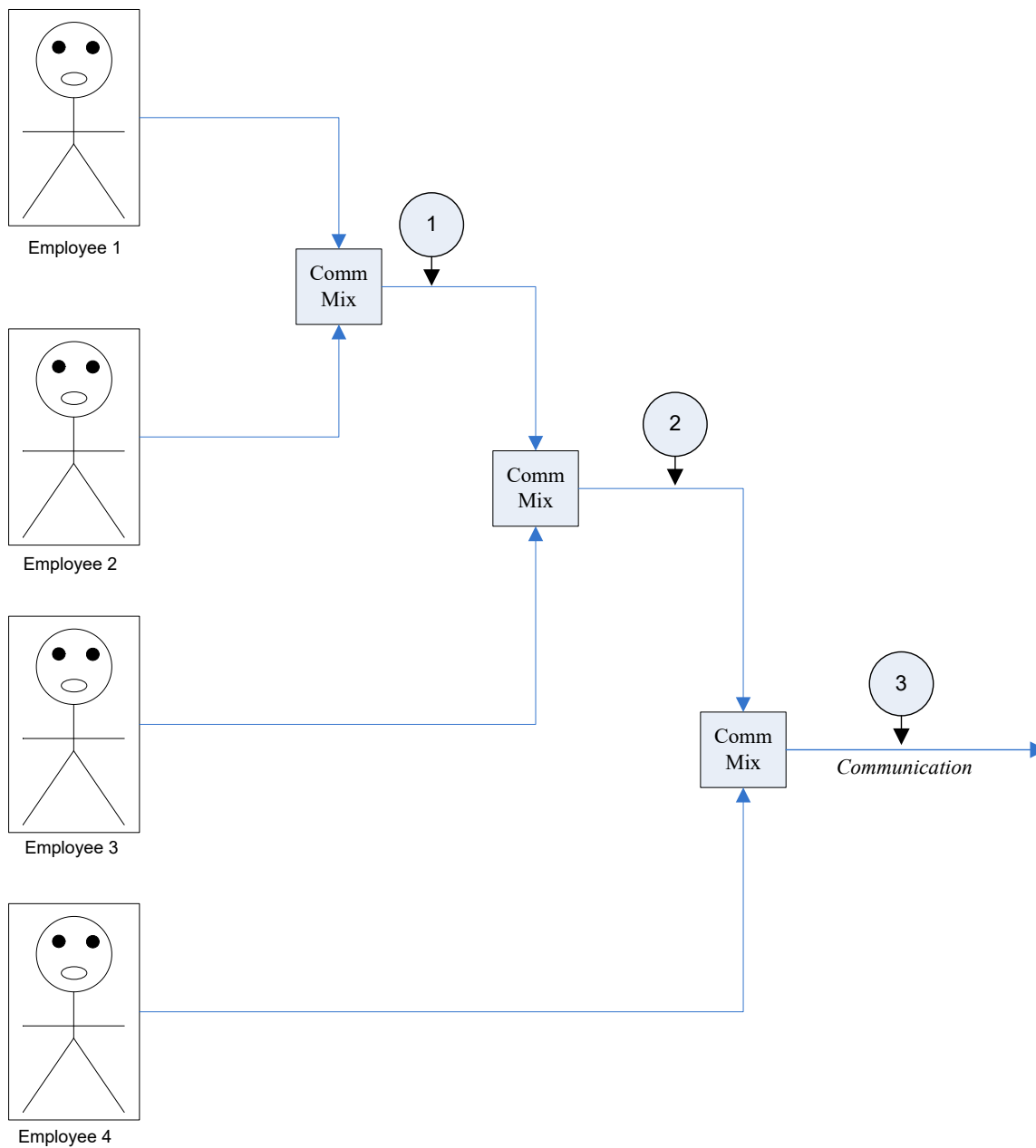
Within a conversation or a communication, we can recall an important part of that communication. Within a communication or a conversation, we can recall an important point

of that communication. We use the callout to show an important point of a communication or a conversation. Assume that two employees are communicating together about a project, if one of the employees record an important point of that conversation, then a callout can be used to show that. Whenever we use the term important point, we use a part of the communication that is important to the project or the employees for instance. In the model below, we use callouts to record important parts of the communication by the employees.



If we don't want to use callouts to show important points of a conversation or communication, we can also use node with numbers to show that. The model below is the same as the one above, except nodes with numbers are being used instead of callouts. When we use nodes with

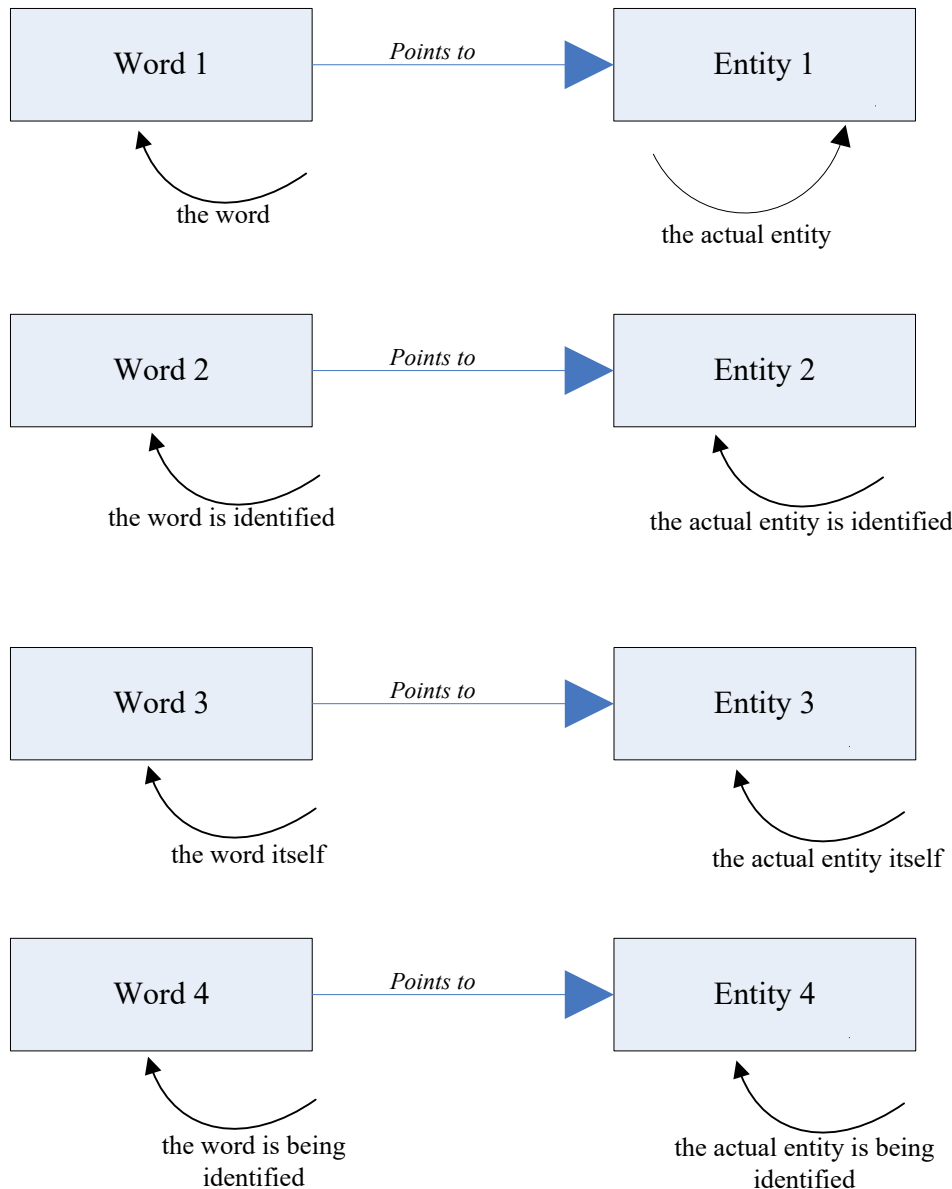
numbers, we can also use a table to provide more information about those nodes. The table below the model provides information about the nodes.

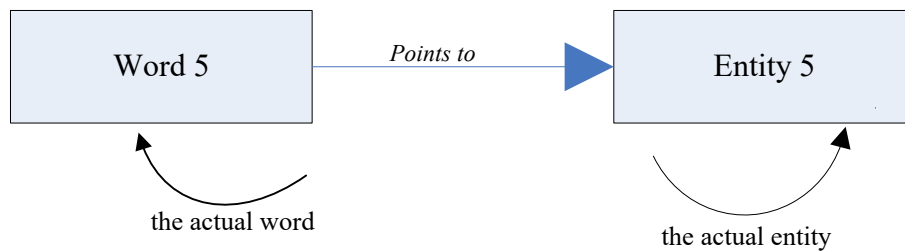


Node Number	1	2	3
Node Information	Employee 1 communicates with employee 2 about an issue	That same communication carryovers to employee 3 about the same issue	The communication continues to employee 4

Example Number 18

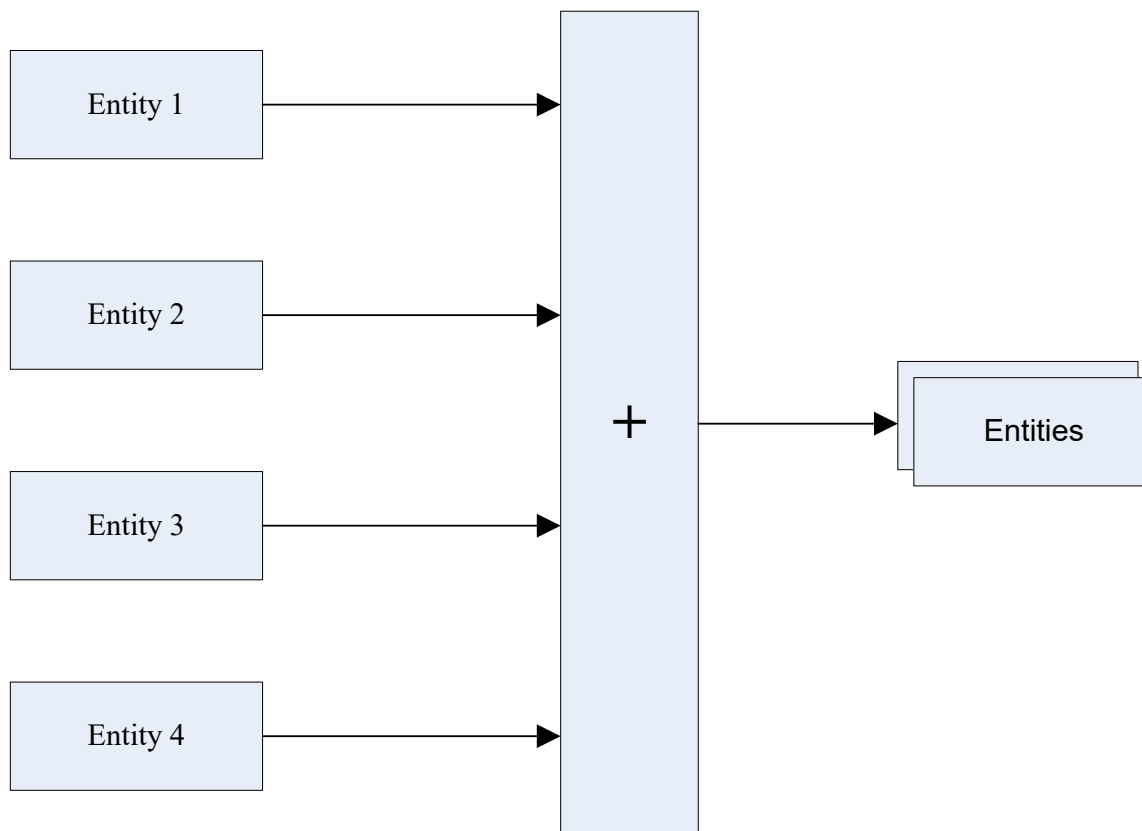
While we use words for oral and written communications, we don't think about them directly. We think about actual entities they represent. For instance, if we use the word *table* we think about an actual table, which is a physical table. If we use the word *tree*, we think about a tree which is a physical entity. The point to label or the point to arrow label enables us to map a word with the actual entity it represents. From the models below, we show the usage of the point to label, where we show each actual entity each word represents.



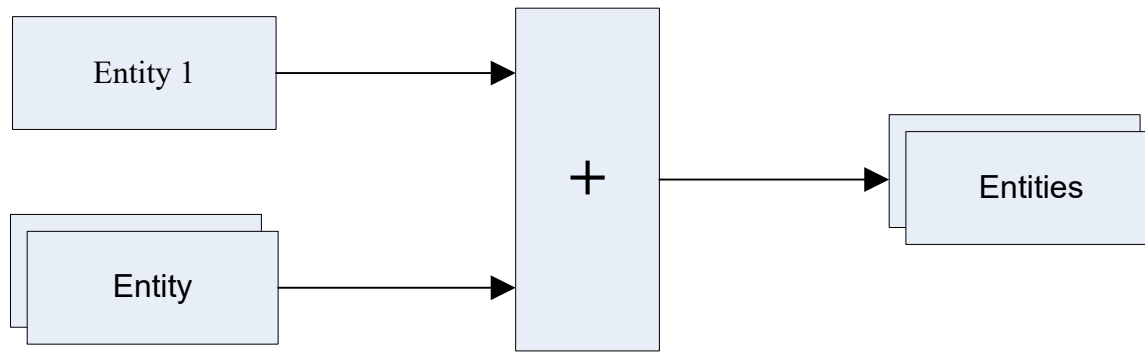


Example Number 19

Within a project, it may be important for us to show actual entities that we use for the project. Assuming that in our project it is required for us to use physical entities, it may be possible for us to show the usage of those entities related to the project. In this case, we can group those entities to show their quantities. The model below shows the usage of entities in our project. We can only do that if it is desired. In the model below, we use the grouping entity to group the entities.

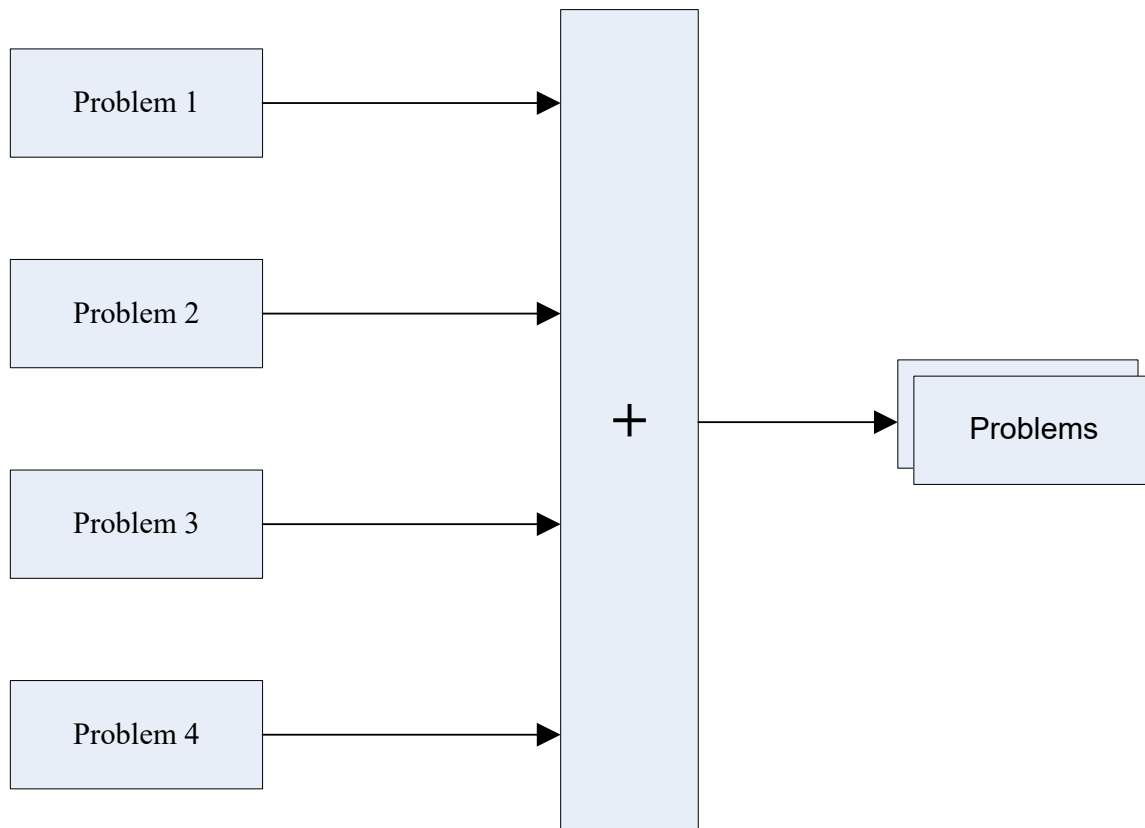


To preserve space, we can also use the grouping entities in the form of one on top each other to show the grouping of the entities. The model below shows a different way of grouping the entities to the left. This model is the same as the one above, except we group the last three entities in the left in order to reduce space.



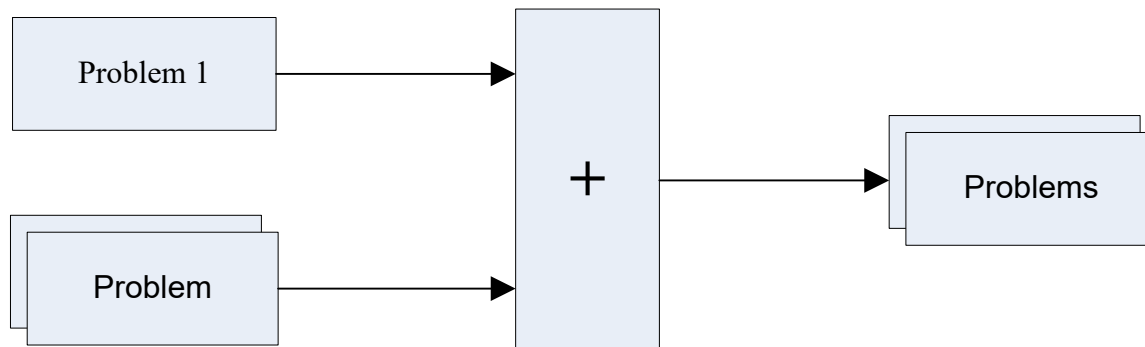
Example Number 20

Since problems are considered to be entities themselves, they can also be grouped. Within our project, it may be possible and require for us to show all the problems we have in the project. By grouping all the problems we have together, we show the total problems we have in the project. The model below shows the grouping of the problem we have in the project. We use the grouping entity to group the problems.



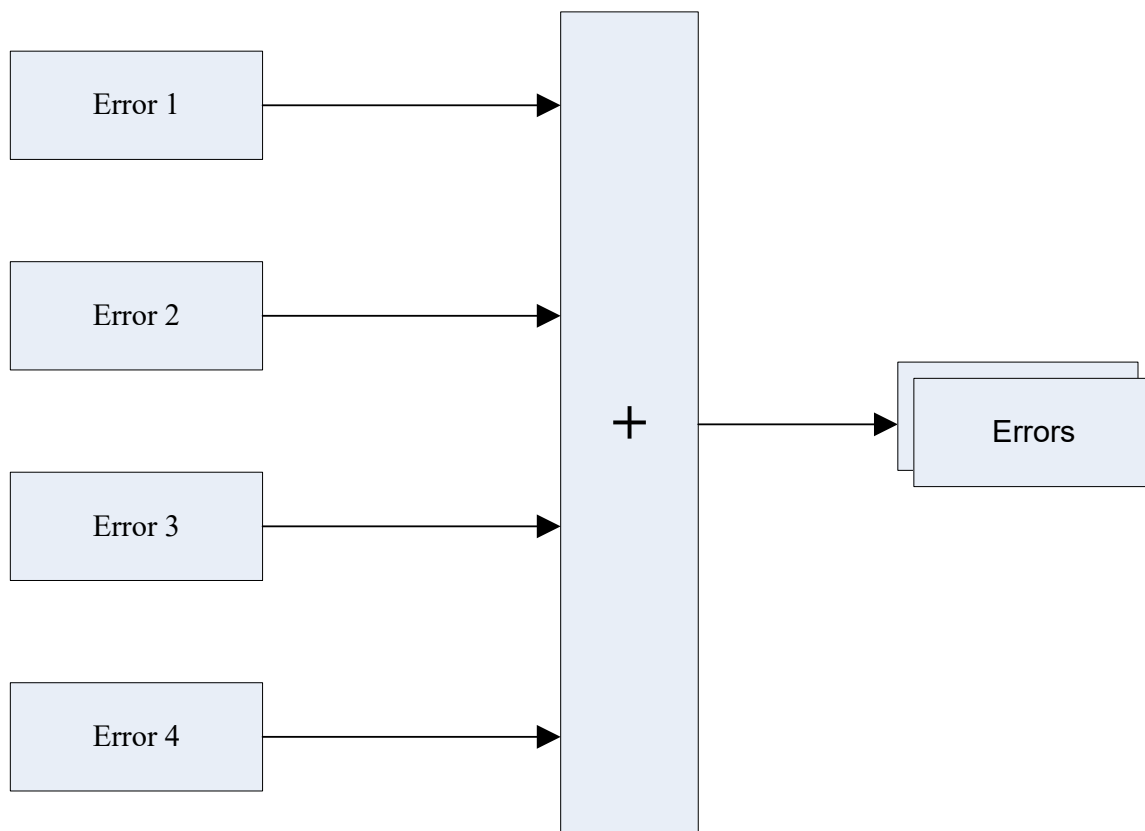
As we did for the entity grouping model, we can also do the same for problems. To preserve space, we have grouped the last four problems in the left above in a form of one on each other.

The result gives us the model below. Disregard the way we group them, the output is always the total problem for the project.

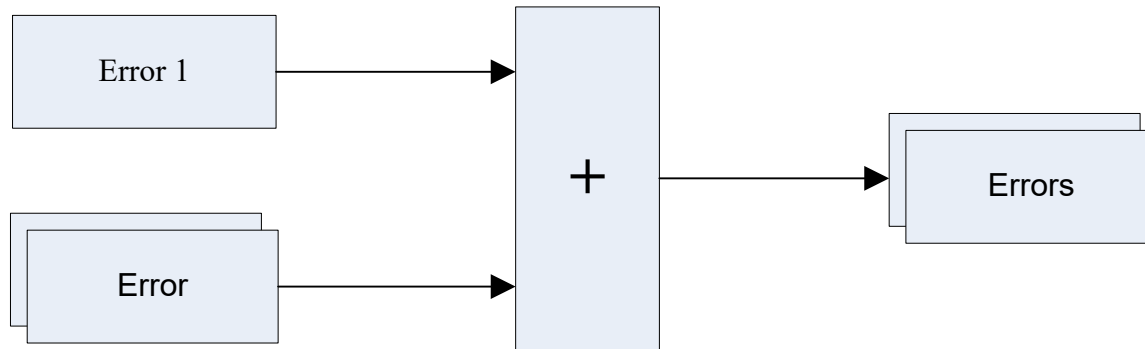


Example Number 21

Since errors in communication gives rise to problems, an error that occurs in a project can develop problems, so it may be important for us to accumulate all errors in our project so we can keep track of them. The model below shows a way for us to group our errors in our project. We use the grouping entity to group the errors.

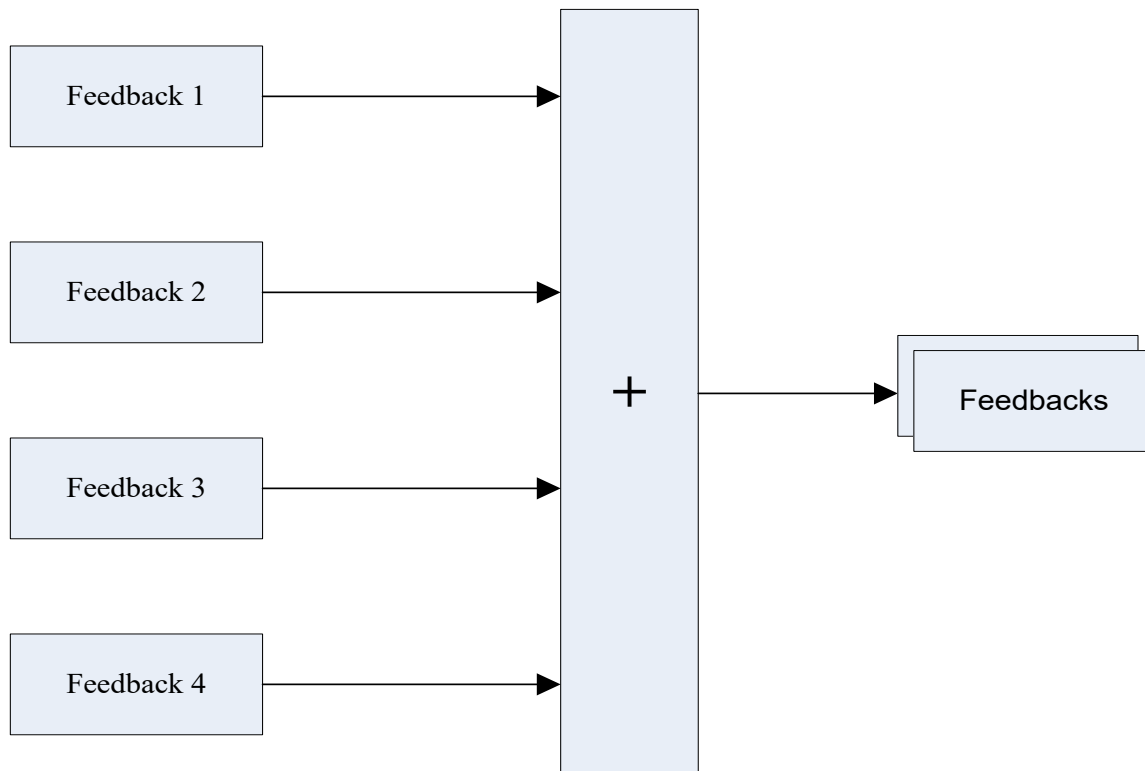


To reduce space, we can also group errors in a form where we can show one on top of each other. The model below shows that form of grouping errors. This model is simply the same as the one above, except the last three errors in the bottom are being grouped to form a single one. The result is still the total error for the project.

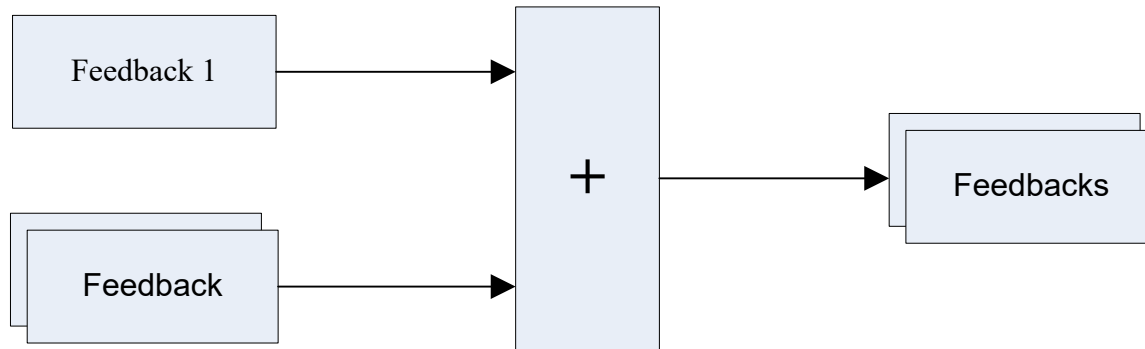


Example Number 22

It may be important for us, in a project, to track all the feedback that we receive for the project. In this case, we can group all the feedbacks so we can keep track of them. By using the grouping entity in the form showing above, we can group all the feedbacks we receive in our project. The model below shows the total feedback of the project.

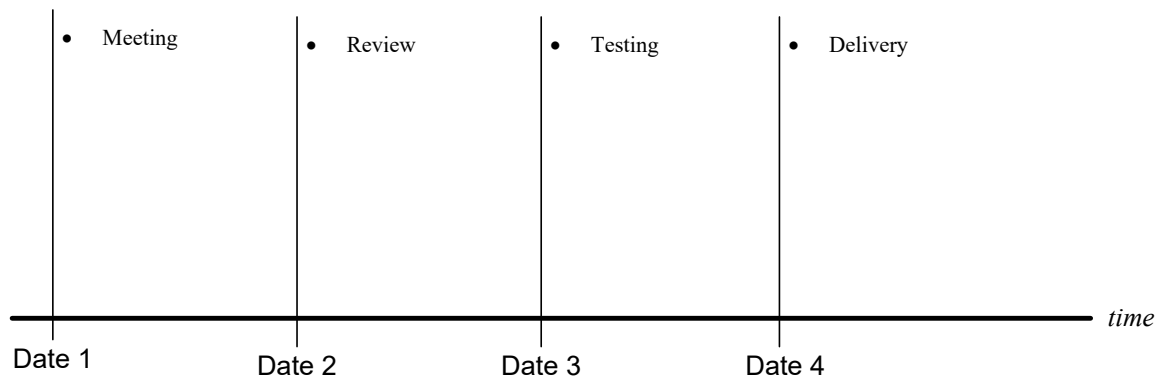


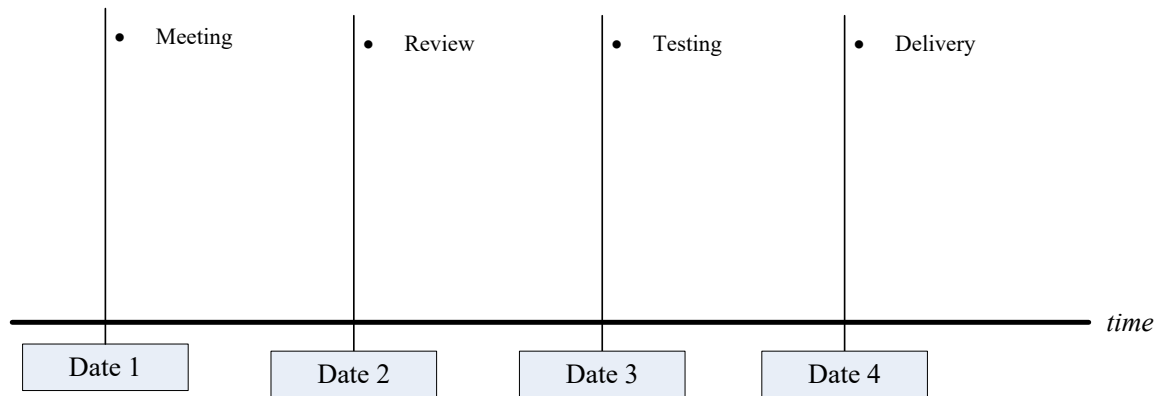
While we use the model below to sum all the feedbacks we have received in the project to reduce space, we can also group them in a form of one over each other. The model below is the same as the one above, except we group the last three feedback one on top of each other. The output still provides the total feedback.



Example Number 23

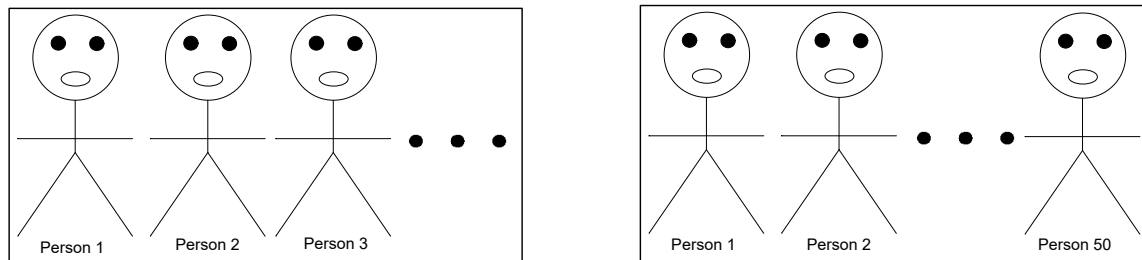
We can use timeline and dateline to show our project timeline and our project schedule. The model below shows the usage of timeline and dateline to show a project schedule. While we use number with the date to show specific date, it does not matter. The date can be any date that is related to the project.





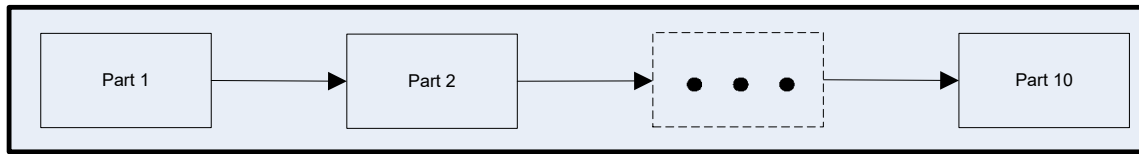
Example Number 24

Assume that we are working on a project, where we have 50 people on our project. Now while modeling our project, it may not be possible for us to show all of them. For instance, assume that we are modeling our project in a piece of paper or a drawing board, for illustration purpose, we may not want to show all of them. In this case, we can use continuity in this form to show the people in our project. For the first model, the continuity means there are some other people after *Person 3*, while in the second model, the continuity means there are some other people between *Person 2* and *Person 50*.



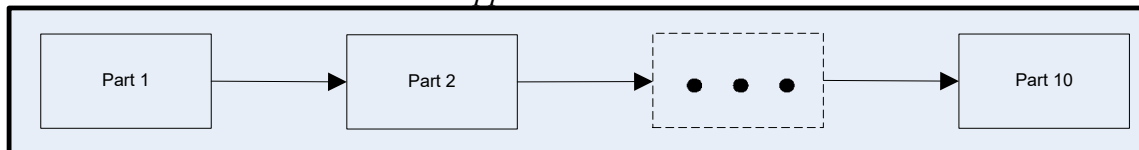
We can also use continuity to show the continuity of parts of our communication function and also the continuity of part of our application. The model below shows what we are talking about. Assume that we are working in a project, where our communication function is made of 10 different functions. In this case, we can show them in the following form. Assume that we have limited space to show all the functions, here we use continuity to show there are some functions between the second one and the last one that are not listed.

Communication Function



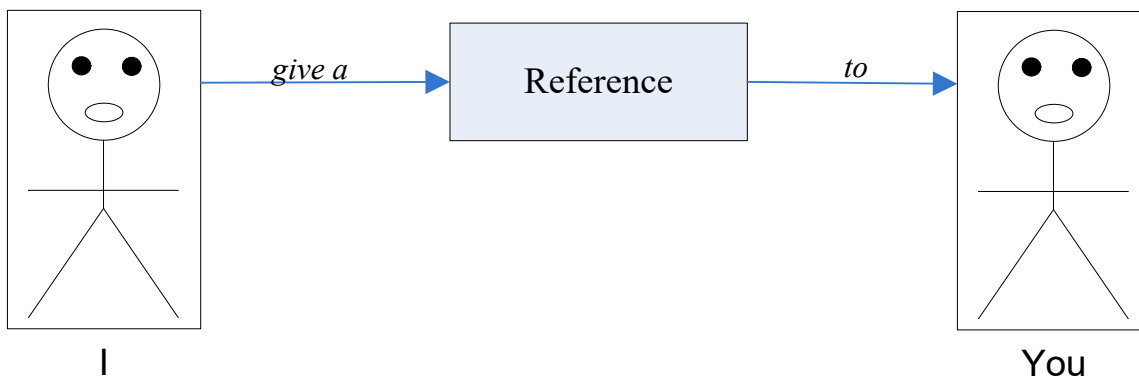
As we did for the communication function above, we can also do for the parts of application and any other entity that can be used with continuity. Assume that we are working in a project, where our application is made of 10 parts; in this case we can use continuity if we don't want to show all of them as shown by the model below.

Application



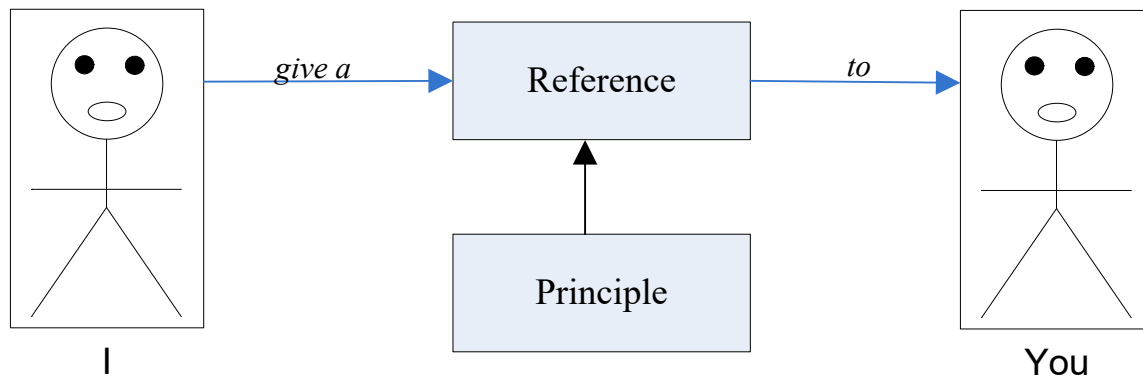
Example Number 25

Since references contain information, it may be possible for references to be given to us at a time we can use it and when it is necessary. The reference entity can be used in the form below to show a reference given from one person to another person.

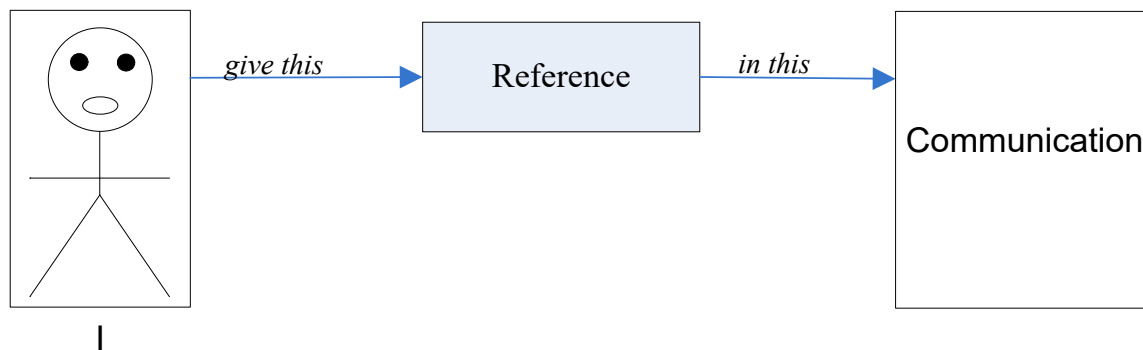


Since not all entities are considered to be references and the principles attached to a reference must be understood in order for a given entity to be considered to be a reference, it may be possible for one to present an entity to others that is not a reference. For that reason, feedback

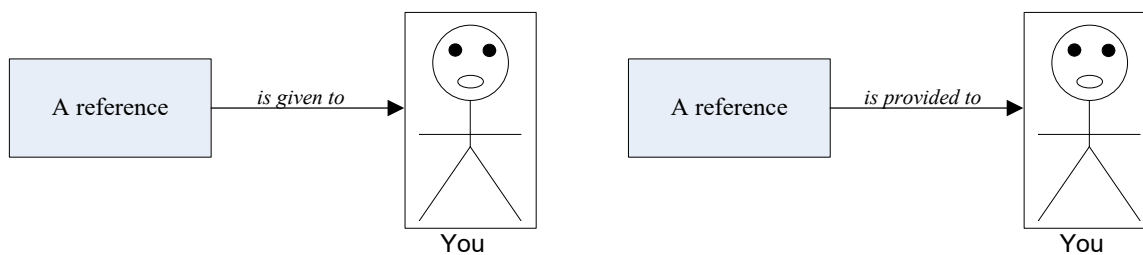
can be used as shown below with the reference entity to determine if that entity is an actual reference.



Since a reference is given in the form of communication, during our communication it is possible for a reference to be given and it is also possible for a reference to be captured. The model below shows that a reference is given during a communication.



Since a reference contains principles, the origin of a reference is not important to us, but the reference itself is what is important to us. For this reason, the one who provides a reference to us is not important to know, but the reference itself is what is important for us to know. In this case, it is good to show that a reference is given without showing the one who provides that reference.

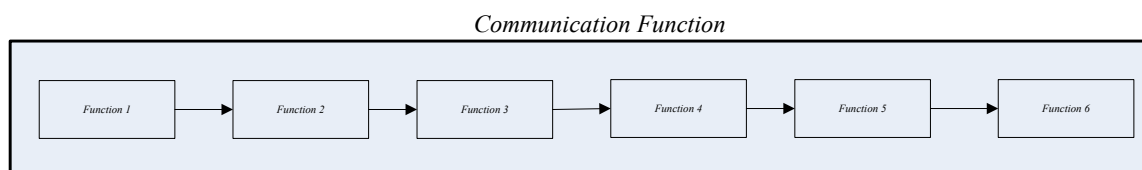


Since a reference is given in the form of communication and the one who provides the reference is not important to know, just the reference itself. In this case, it is possible for us to show that the reference is given in a communication without showing the one who provides that reference as shown by the model below.

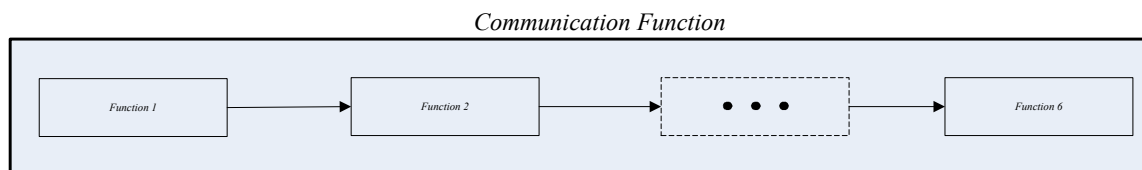


Example Number 26

We use the continuity entity to show more entities that are not visible in the model. For instance, if our communication function includes many functions, it is possible for us to use the continuity entity to show more functions that are not listed in the model. Let's assume that our communication function is made of 6 functions as shown by the model below.

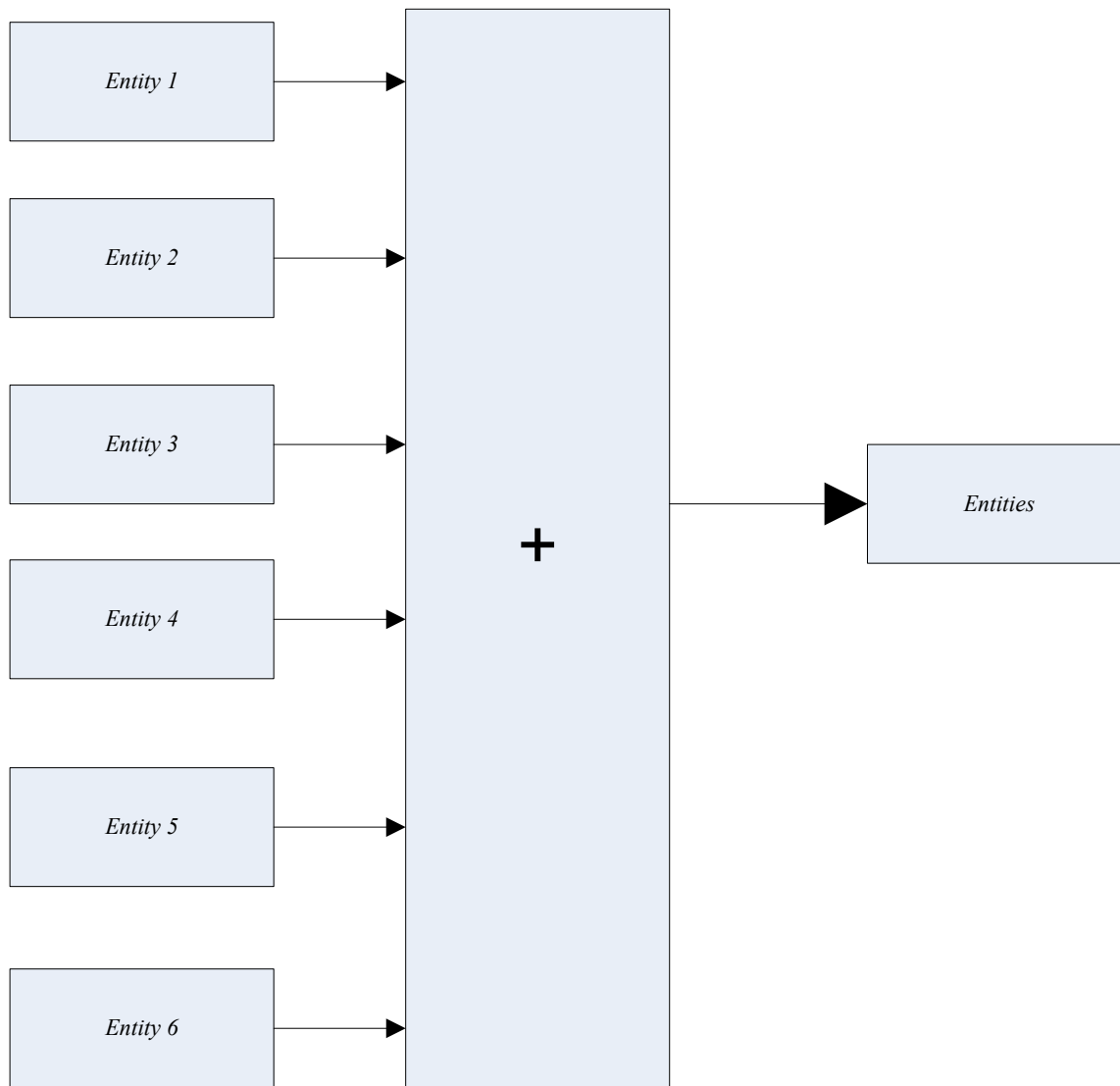


Since our communication function is made of several functions, while modeling our application, it may be possible for us not to show all the functions that include in our communication function. While the model above shows 6 functions, it is possible for us to use continuity to show some of those functions that are not in the model as shown below.

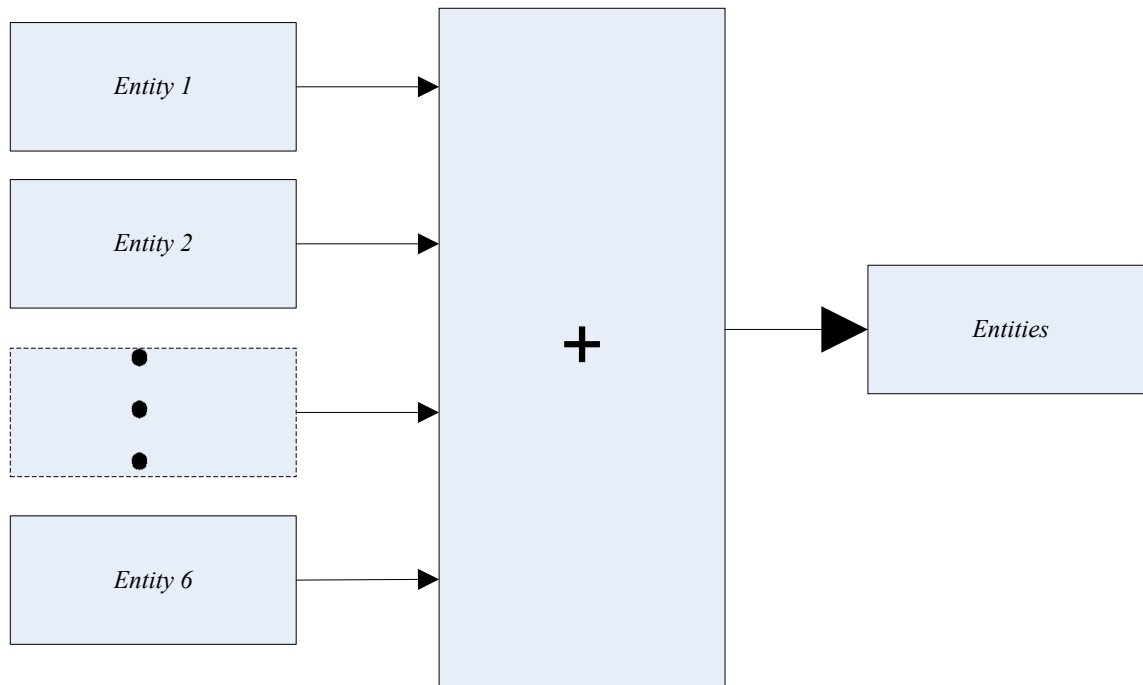


Example Number 27

The continuity entity can also be used to show list of entities in our application. For instance assume that we want to show a list of entities and some of those entities are not visible in the model, we can use the continuity entity to show that. Let's assume that within our application, we have a list of entities that is made of 6 entities as shown by this model.

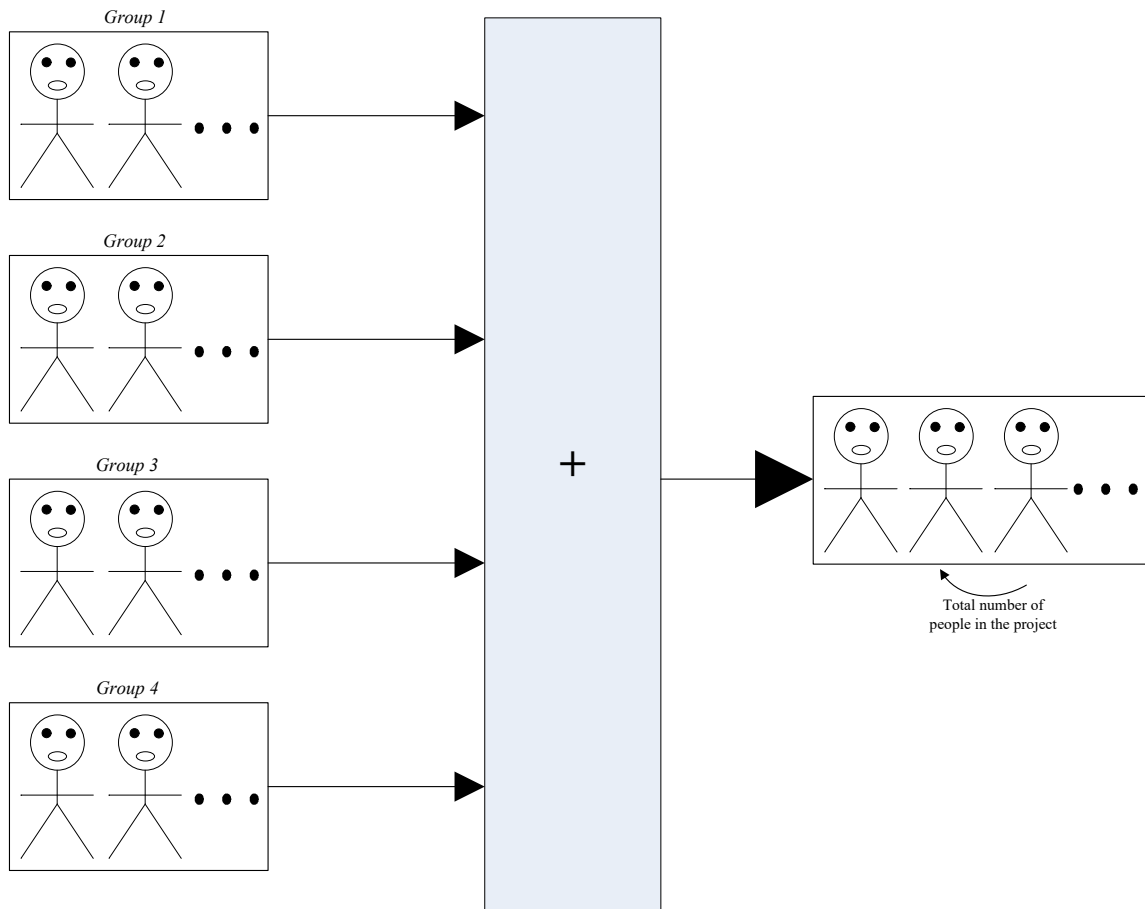


Let's assume that within our model it is not possible for us to show all the entities, and then we can use the continuity entity to show entities that are not visible as shown by this model. The usage of the continuity entity from the model above simply shows a list of entities that are not visible in the model. While we use the continuity entity before the last entity, we can also put the continuity at the end. Putting the continuity at the end shows that we don't know the end of the list.



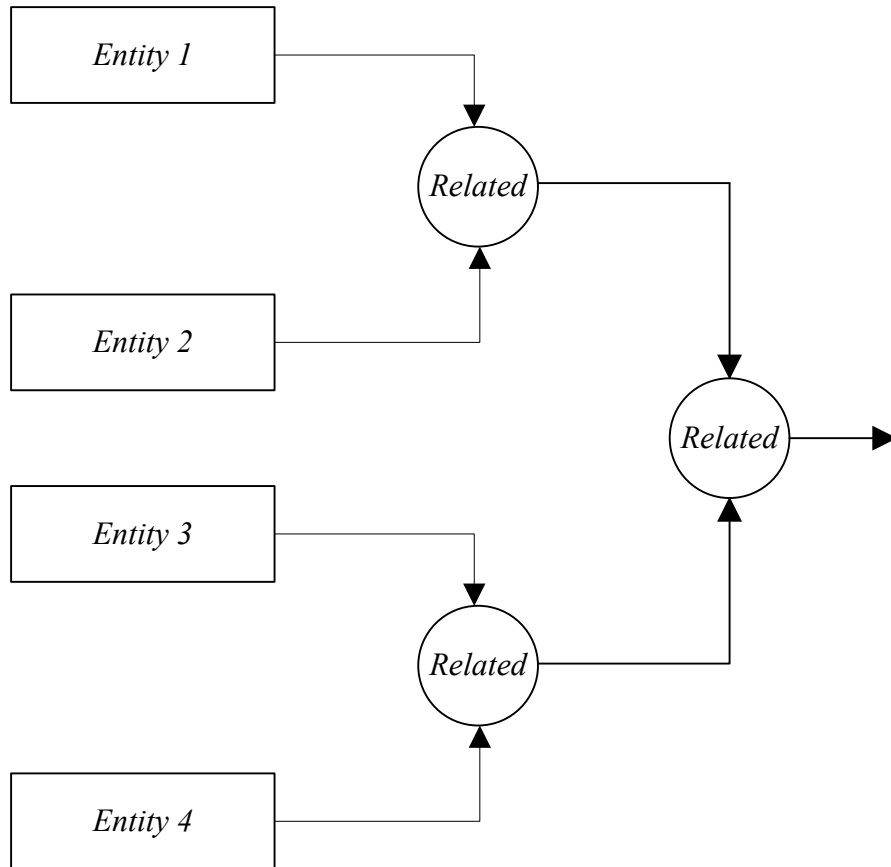
Example Number 27

In a project it is possible to have groups of people that have specific functions. We can use the group of people entity to show groups of people that make up our project. For instance assume that in a project, we have four groups of people. In this case, we can represent those groups of people by the model below. As shown by the model below, our project is made of 4 groups of people. The group to the right shows the total number of people in the project.



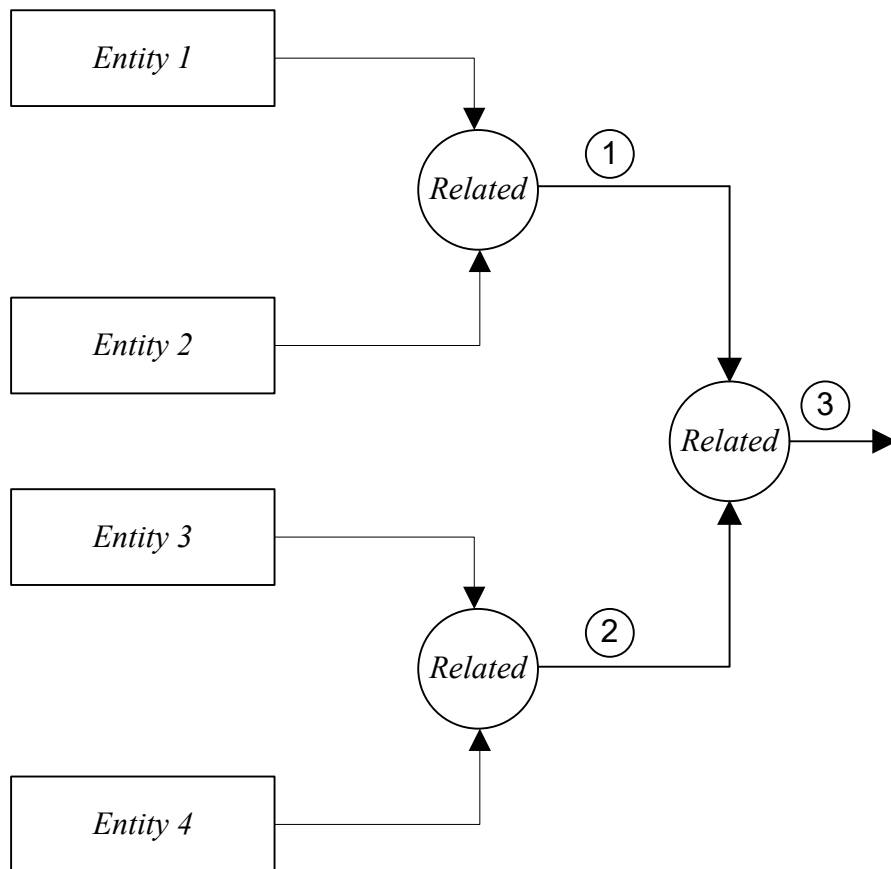
Example Number 28

Let's assume that we have a relationship model that shows the relationship between *entity 1*, *entity 2*, *entity 3*, and *entity 4* as shown by the model below.



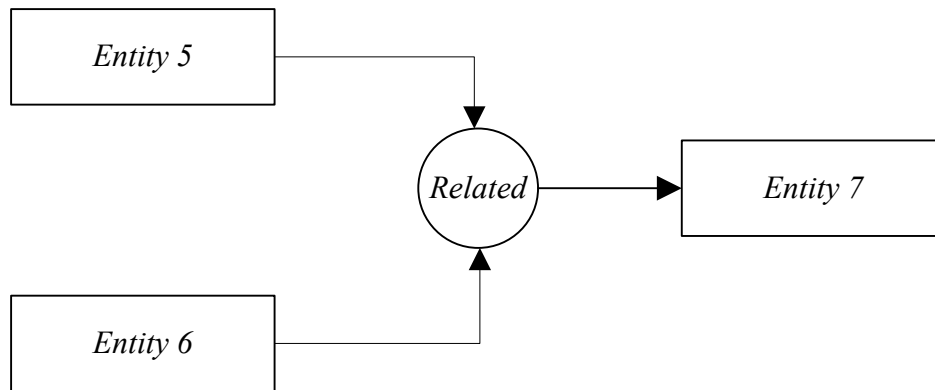
While we draw the relationship above to show multiple relationships or a relationship entity for 4 related entities; when having multiple entities in multiple relationships or multiple entities in a relationship, it is possible for us to define an entity at specific node. Since two entities are related by an entity, and that entity is considered to be the result of that relationship, it is possible for us to identify an entity at a node or identify it by a node as shown by the model below.

Let's assume that *entity 1* is related to *entity 2*, where that relationship is related to the relationship of *entity 3* related to *entity 4* as shown by the model above. To better understand the explanation, let's show that model above again with node number.



From the model above, we redraw the relationship again to show node number. In this case, we have node 1 that shows the relationship between *entity 1* and *entity 2*, node 2 that shows the relationship of *entity 3* and *entity 4* and node 3 that shows the result of the relationship from node 1 and node 2. In this case, node 3 shows the relationship of the 2 relationships.

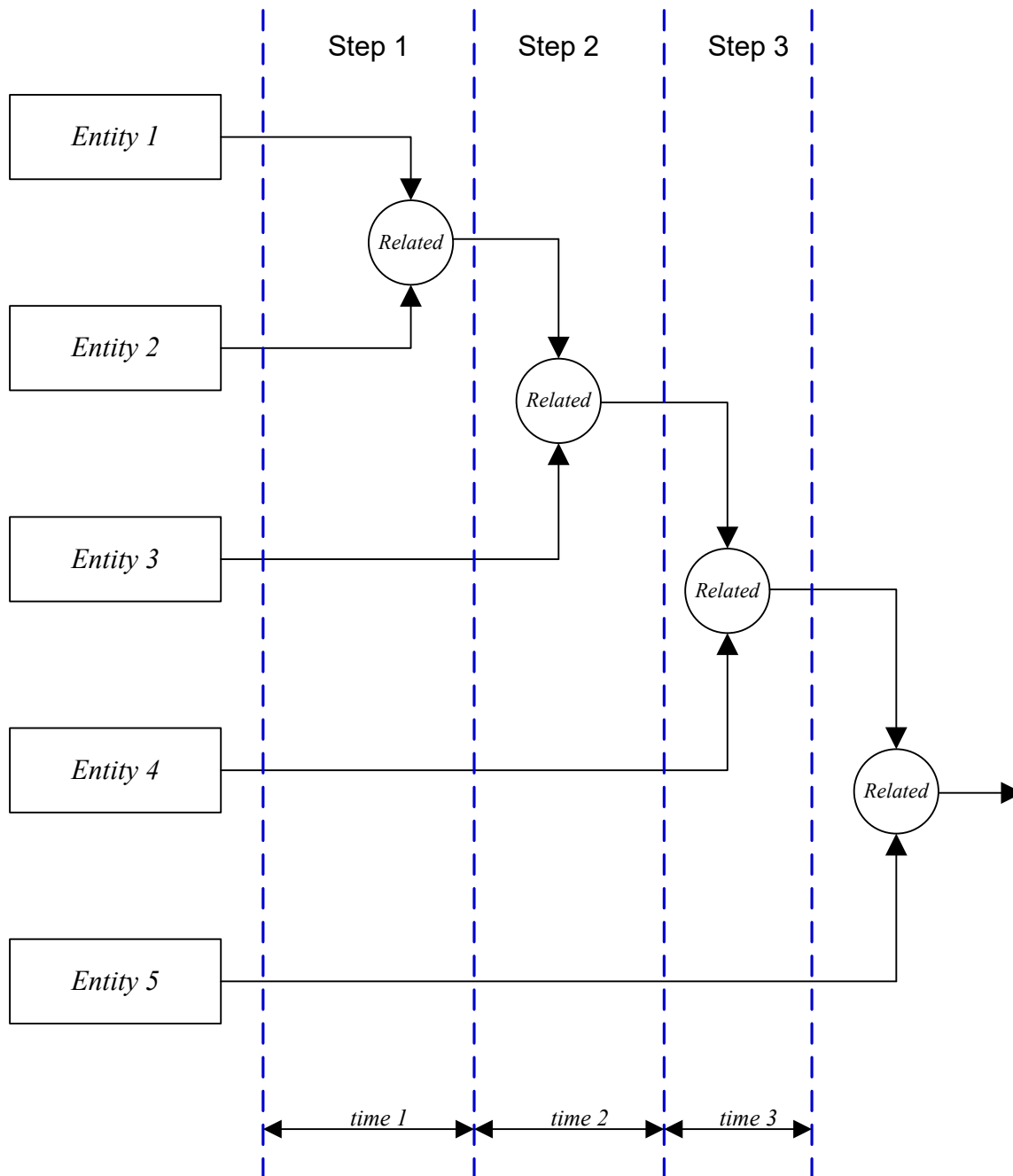
Now what we can do is we can identify entities for node 1, node 2, and node 3. Assuming that we can identify the entity, entity 1 and entity 2 are related by, and then we can define that entity. Here, let's define or identify that entity as entity 5. Then we can identify the entity, entity 3 and entity 4 are related by entity 6, then we can identify the entity those two relationships are related as entity 7. What we have done here is we have simply identified node 1 as entity 5, node 2 as entity 6, and node 3 as entity 7. In this case, we can redraw the relationship model in the form below with those entities.



As we can see, if we can identify a specific entity at a specific node in a relationship, then we can reduce the number of entities in that relationship. Since there is no limit in a relationship in term of number, this process has no limit.

Example Number 29

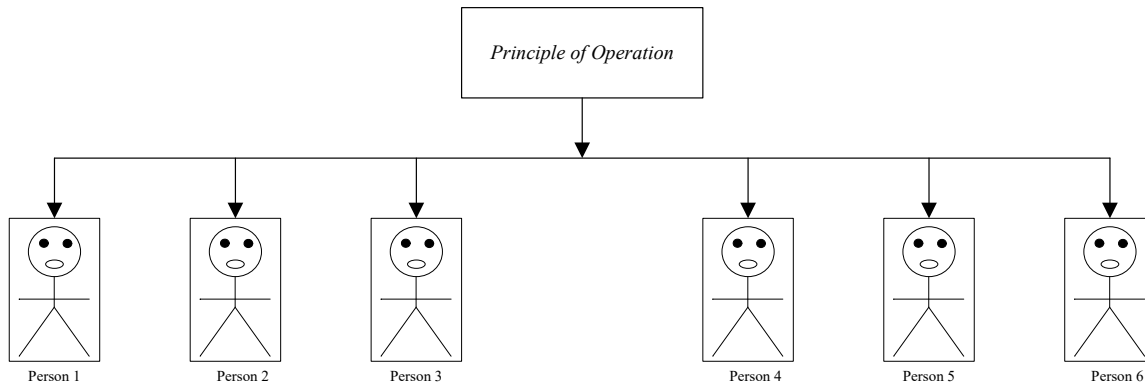
Since it is more understandable for us to approach relationships one at a time, the separation line and the time line can be used together in a relationship model to show the way we approach or identify a relationship at a specific time. In other words, since it is better for us to identify the relationship of two entities at a time, rather than several entities at a given time. In this case, we can use the separation line with time line to show how we approach or identify a relationship at a given time. For instance, in relation to the model below, we identify relationship one which is the relationship of entity 1 and entity 2 and then later at time 2, we identify the result of that relationship in relationship with entity number 3 and so forth.



Example Number 30

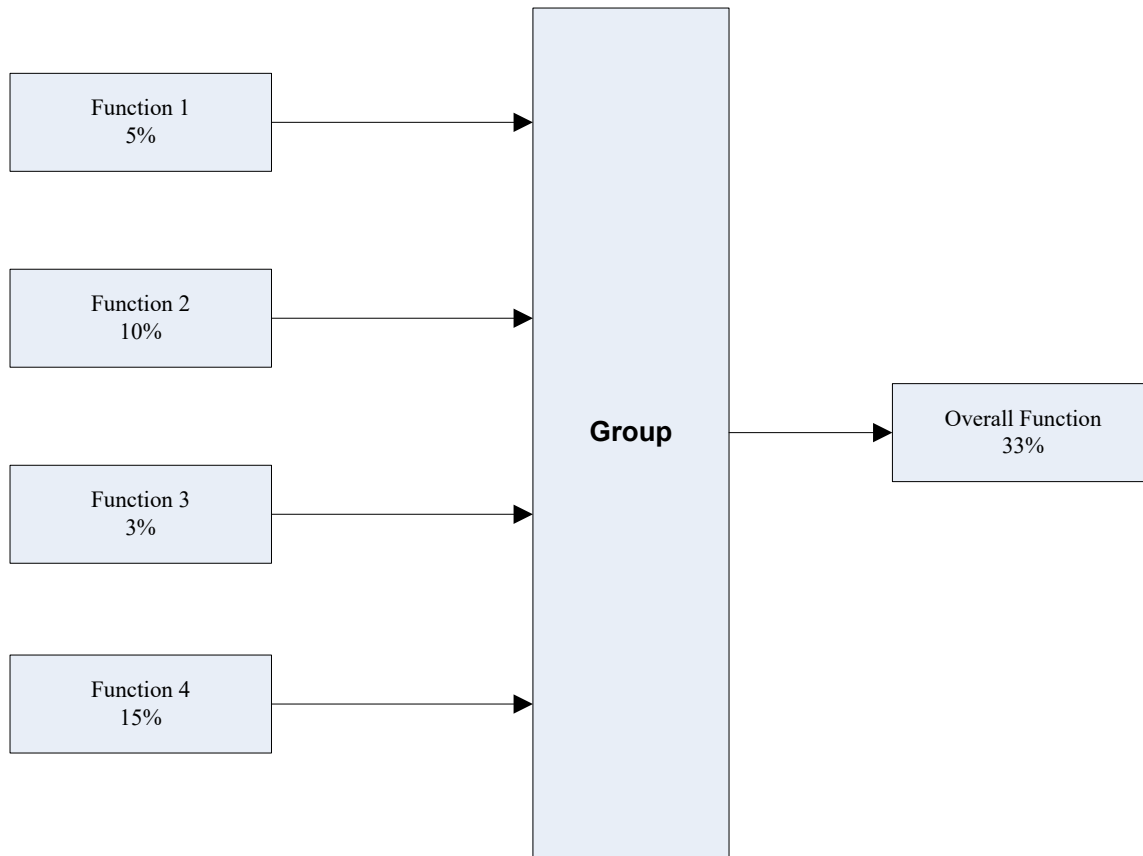
Since our application depends on our understanding of the principle entity. By having an operating principle, it is possible for us to follow that principle in order for us to execute our project. Given that the application itself depends on everybody in the project, it makes sense for those people to follow the operating principle to enable the execution of the application. Within the application itself, it makes sense for us to show everybody in the application related to the principle of operation as shown by the model below. The model below shows that we have six

people in our project. It does not matter in term of number of people we have, we can still show them with the same operating principle. We can also use continuity and groups if space is an issue.



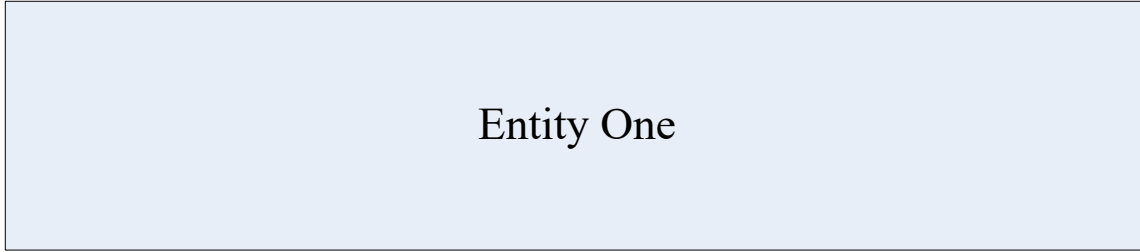
Example Number 31

Given that within the project itself, the status of a function is considered to be an entity; similarity in the list of entity models, if necessary, it may be possible for us to show the status of each function that makes of our application or the overall function as shown by the model below. In this case, we can use the function name or the actual function and the completion status of the function; for instance if our application or the overall function of our application is made of 4 functions; then it is possible for us to show the status model in the form below. The model below shows the % completion of each function and the overall % completion of the overall application.

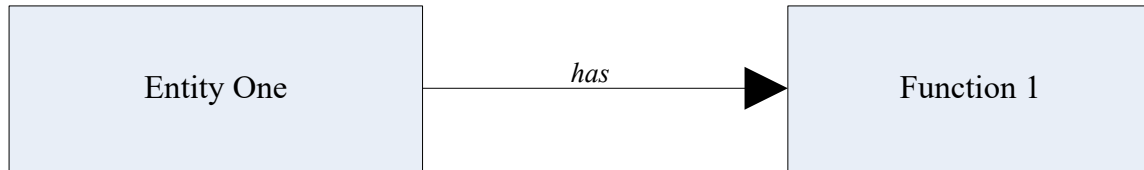


Example Number 32

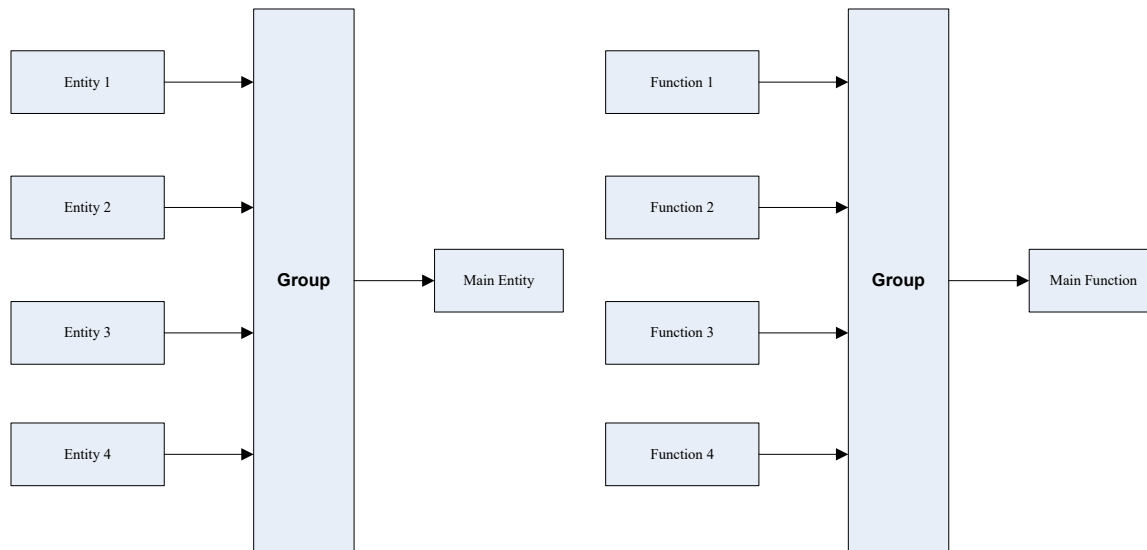
The overall process of modeling enables us to model our application, disregarding the type of that application. For instance, if our application is being viewed as a service that we provide, it is possible for us to model it. As well as, if our application is being viewed as a product that we develop, it is possible as well for us to model it in the same manner. In term of modeling our application, if the result of our application is being viewed as a service, that service is being viewed as an entity. As well as, if the result of our application is being viewed as a product that we make, that product is being viewed as an entity. By understanding that, in terms of entity, it is possible for us to show or identify that entity within the model of our application. For instance, if our application is to produce a physical entity, then we can show that entity produced by our project or our application in the form below. We can call it any name or its actual name, here we simply call it *Entity One*.



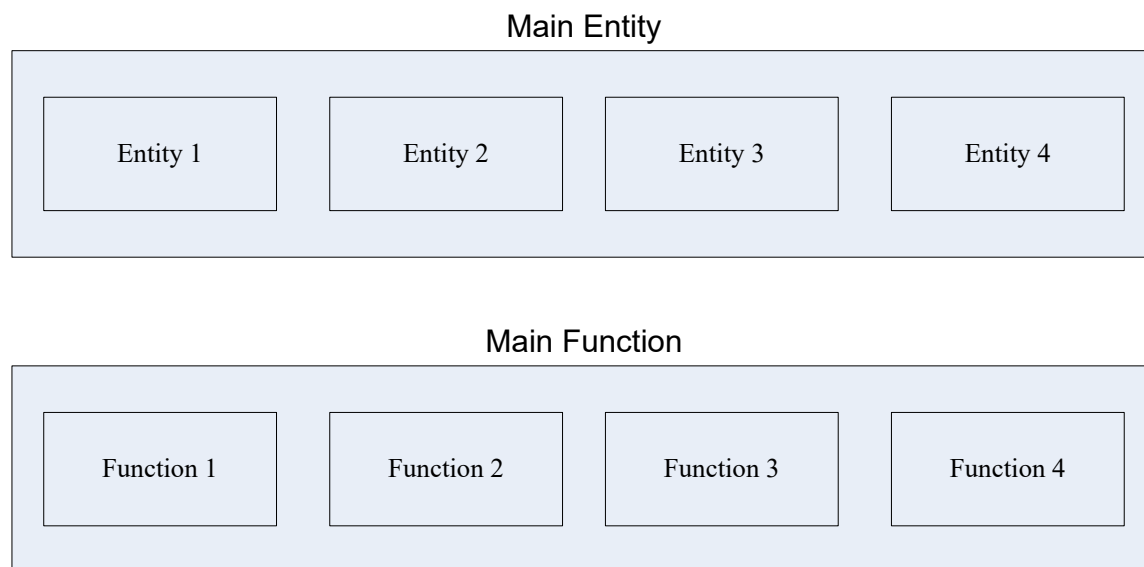
From the model above, *Entity One* is being viewed as the entity produced by our application or our project. If the purpose of our project is to provide a service or service an entity, then *Entity One* is being viewed as well as the service we provide. Since an entity must have a function and the function of an entity is also an entity, then it is possible for us to show *Entity One* and its function in the form below. Here we use the name *Function 1* as the name of the function of the entity; we can use any name we want or the actual name of that function.



The models above show the entity produced by our application and the function of that entity. Given that an entity can have several parts and each part of an entity can be a function within our application, it may be possible as well to extend the models above to the form below. From the models below, the first one to the left shows a list of entities or a list of the parts of an entity produced by our application, while the one to the right shows the function of each entity or each part of the entity.

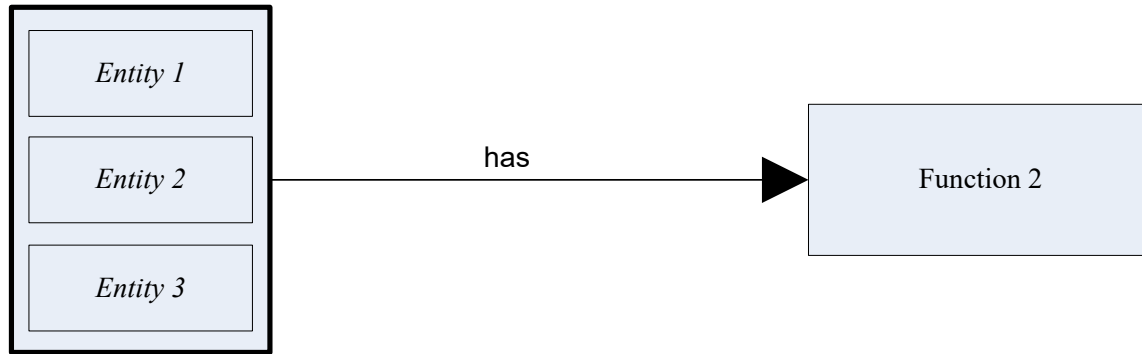


From the models above, the main entity is being viewed as the entity produced by our application. While in the second model, the main function is being viewed as the function of the main entity. Since the main entity is made of several parts, if we want to, we can show it as a group of entities or a group of parts of entities, and so does the main function. While we show both models in the form above, we can also show them in the other form below.



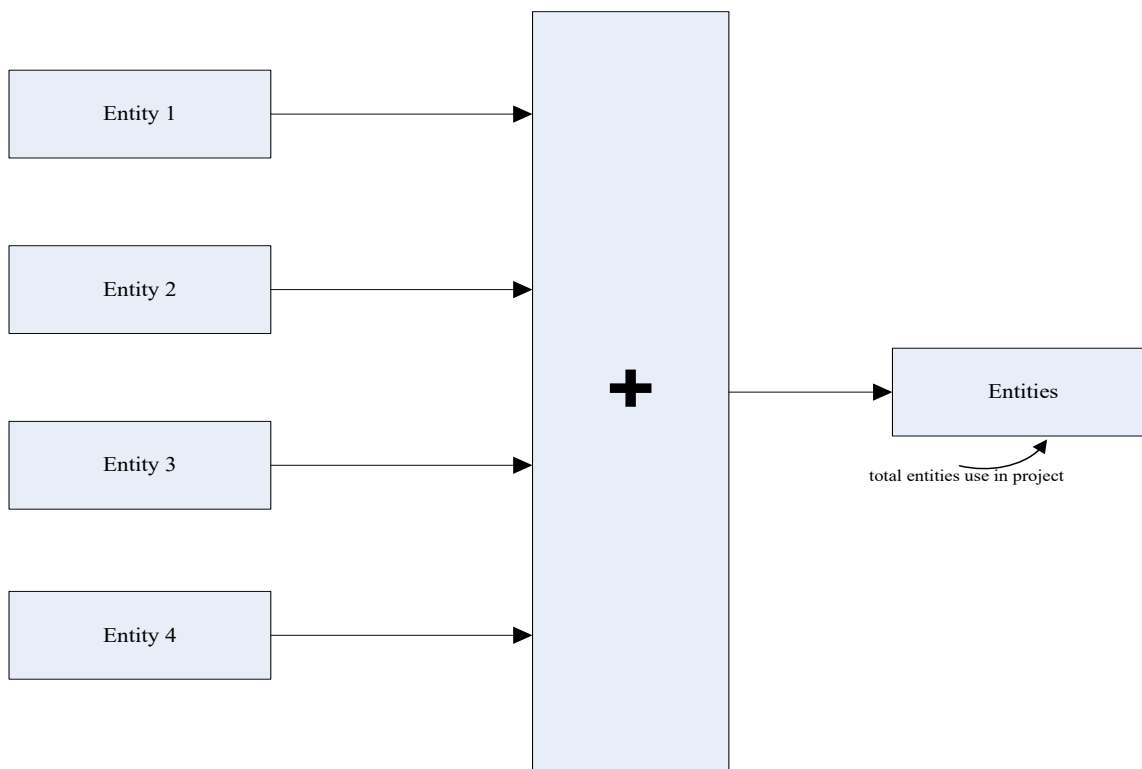
Related to the model above, since entities that we use in our application must have functions, in this case we can use several entities in our application to execute a function. The way to look at it, several entities in our application can have one or more function. Assume that *Entity 1*, *Entity 2*, and *Entity 3* in our application have *Function 2*. In this case, we can show that by the model

below.



Example Number 33

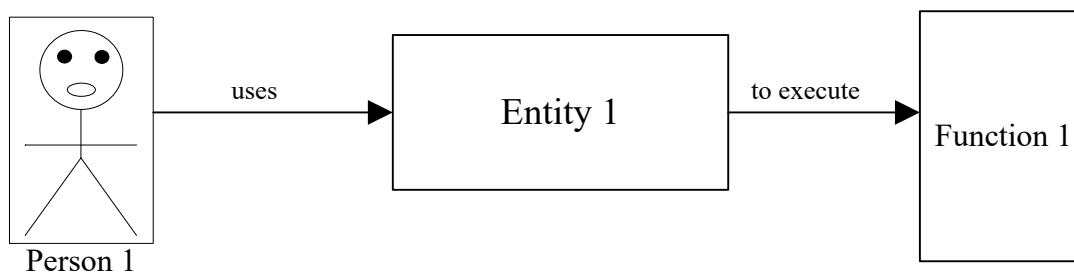
In an application or project, it may be required for us to use many entities in order for us to execute the function of that application. Within the project itself, if we use entities or external entities to help us execute our function, it makes sense for us to show the list of entities that we use in the project. The model below shows a list of entities that we use to help us execute the function of our application.



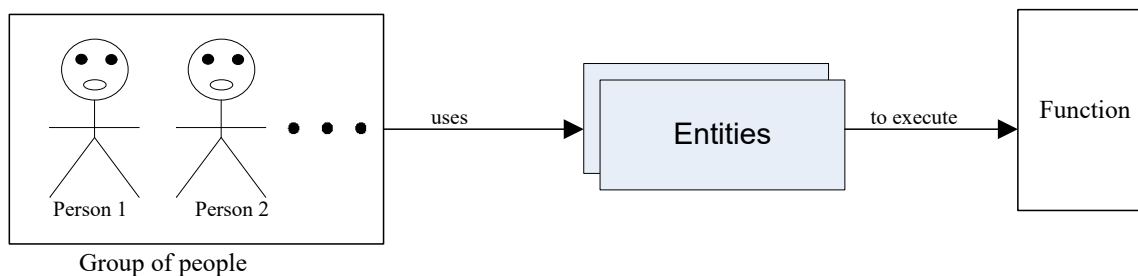
While we use the model above to show the list of entities we use in our project, we don't have to show it in that form. We can also show in any other form.

Example Number 34

From the above example, we have shown the list of entities that are used in our application. In this case, we show the list of entities that we use to enable us to execute the function of our application. Since we use those entities to help us execute our project, it makes sense for us to show that in a model, where we can show that the people in the application actually use those entities to execute the application. By doing so, we can show that in the form presented below. In the first model, we show that *Person 1* in the application uses *Entity 1* to execute *Function 1*. In this case, *Function 1* is considered to be the function of *Person 1* in the application.



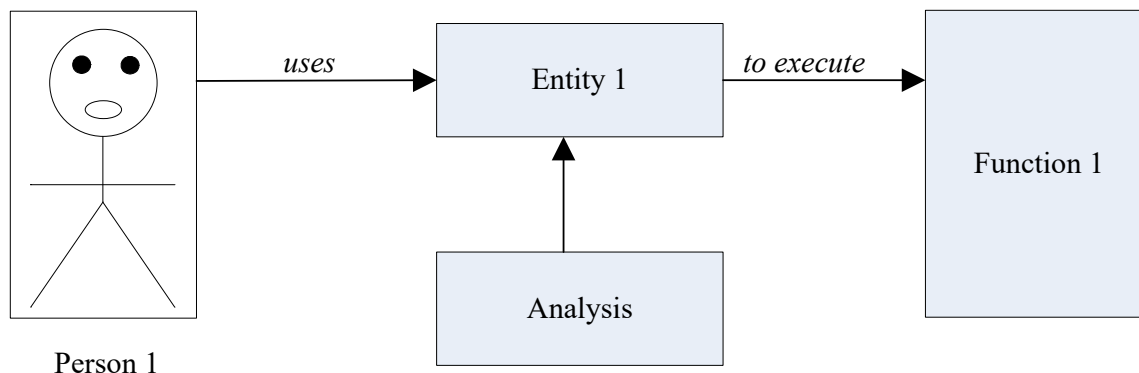
If our application has more than one person and in this case our application has many people, it makes sense to show the entities they use to execute the application. By doing so, we can say that a group of people use a group of entities to execute the application or the function of the application. In this case, we have the model below. The function to the right is considered to be the function of our application.



Example Number 35

By understanding the example above, we can see that the principle entity or the analysis entity can be used to show the analysis of an entity that we use to execute a function. Assume that we

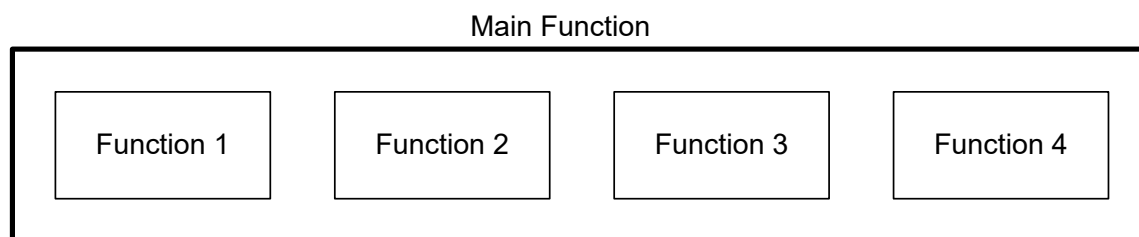
are working on a project where we use an entity to execute a function; in this case we can use the analysis entity in conjunction with that entity to show that analysis of that entity as shown by the model below.



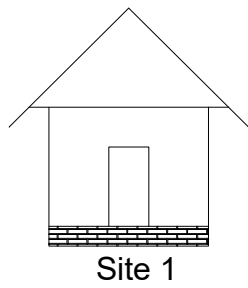
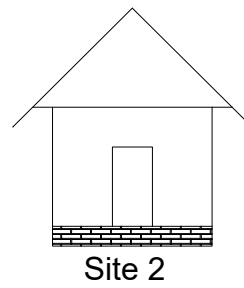
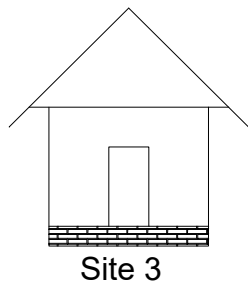
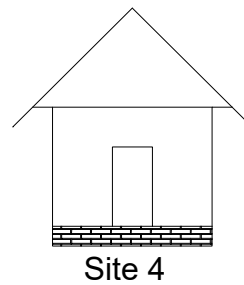
From the model above, we can see that *Entity 1* which is used by *Person 1* to execute *Function 1* is under analysis. From the same model, *Function 1* may well be the function of *Entity 1*, so it makes sense to analyze *Entity 1* related to *Function 1* in that application.

Example Number 36

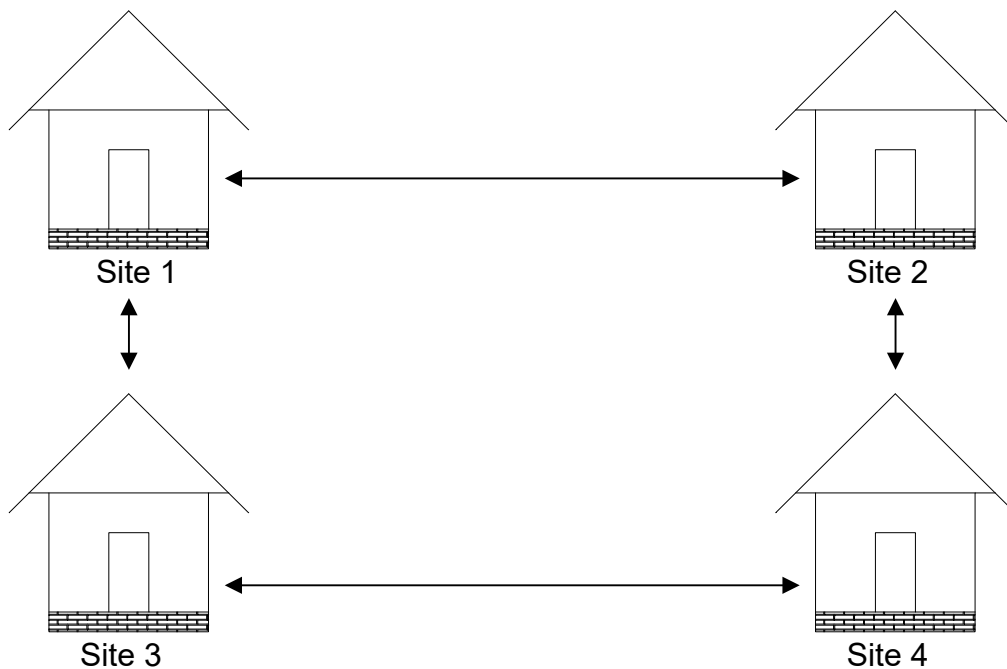
To better understand the usage of the location entity, let's assume that our application includes many people, where those people operate at separate locations while working together to execute the functions of our application. In this case, in each location, people who work in that application have their own functions in terms of groups. To better understand that, let's assume we have 4 groups of people where they work at separate locations. Below the main function of the application is related to the functions of the groups in the form of



From the model above, the main function is considered the overall function of our application, where *Function 1* is considered the function of *Group 1*; *Function 2* is considered the function of *Group 2* etc. Here the overall function of the main application is made of 5 functions, where each function is considered to be part of the application function. In this case if we want to, it is possible for us to show the locations or the sites of operation of those groups of people in the form below.



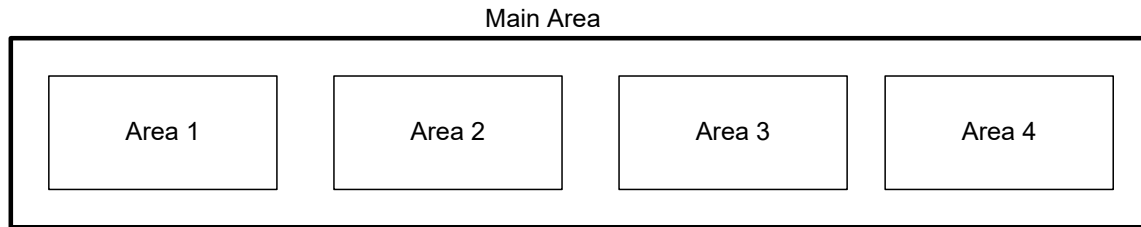
Since the people in the sites or at the locations communicate together to execute the function of the application, if we want to we can show the connection of the sites in the form shown below.



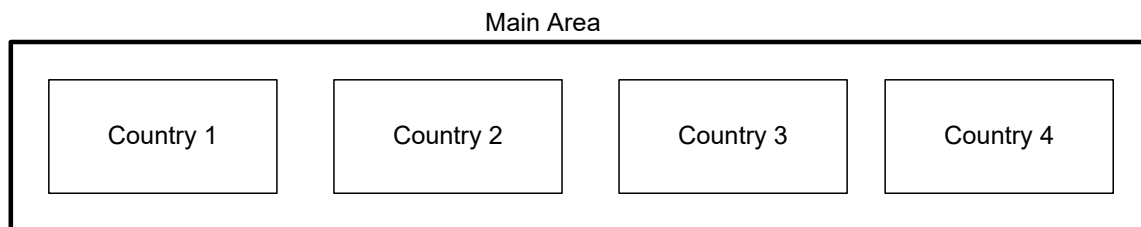
Example Number 37

By understanding our parent principle, it is possible for us to operate at multiple locations. Since our application should not take location into consideration, our understanding of the principle enables us to operate at multiple places, where we do not take specific place into consideration. Let's assume that we operate in 4 areas, then we can use the area entity to show the places that

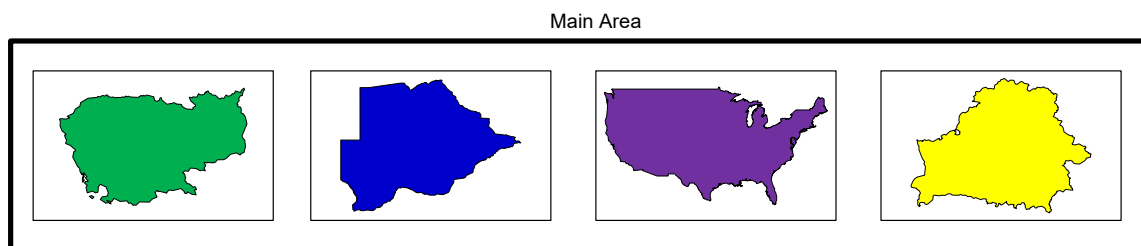
we operate as shown by the model below.



The areas that we operate can be viewed as entities; they can also be viewed as countries as well. We can use the area entity to show that we operate in multiple countries or cities. If we want to, we can use the main area entity to show the cities or the countries that we operate in the form below. The usage of the form below assumes that we understand our parent principles and that makes is possible for us to operate in multiple countries. In this case we can show those countries in this form



Since our understanding of our parent principles does not limit our mobility, with that understanding, it is possible for us to extend our services to other locations where problems are identified and need to be solved by our services. In this case, related to the model above, we can show the sketches of the actual countries in which we are operating in the form below.

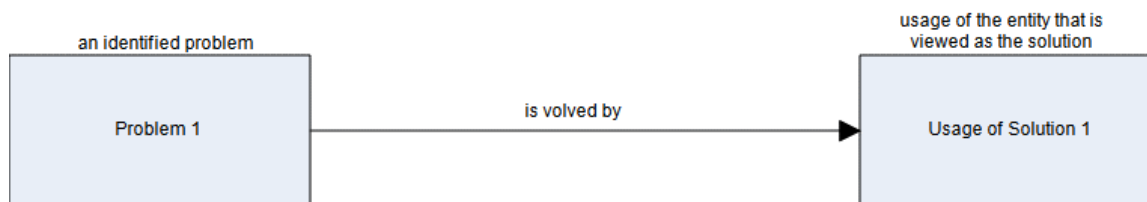


Example Number 38

The problem entity can be used with the solution entity to show the solution of an identified problem. Once we identify a problem, we can use the solution entity with that problem to show that solution of that problem.



While the solution entity is used to show the solution of a problem, a lot of time the solution entity needs to be used in order to solve the underlying problem. It always depends on the problem statement. Considered the practical example in the next section. For example, the *lemonade* is viewed as the solution, however the lemonade itself is not the solution. The *lemonade* must be used to provide the solution. Refer to the practical example for more information. From what we have just said, we can see a lot of time, the model above is represented by the one below in order to be solved by the solution.



From the model above, we can see that the solution itself as an entity may not have solved the problem. In order to solve the problem, the solution as an entity must be used.

Conclusion

It is very important for us to model our application and understand the process of modeling our project in the communication domain. Since communication enables us to do what we do, and we depend on communication to do our work, it is very important to use communication to model what we do. It is also important not to take communication for granted within what we do.

When we model our project in the communication domain, we look at what we do from the flow of our communication. In this case, we can detect and correct errors in a costless manner before they happen. Now that we finish our project and verify that everything is working accordingly, we can then present it or deliver it to our customers. Since communication is not limited by us, since we are not limited by our communication, after delivering or presenting the project to our customers, they may still have questions for us that we need to answer. Now that we understand the process of modeling our project in the communication domain, we should not limit ourselves from communication.

From the paragraph above, we have learned that we are not limited to communication and our customer can question us about the project after being delivered or presented. Now assume that our customers have provided us with two hundred hours to complete the project. We use the time to complete the project and we deliver it to the customers. Upon delivering the project to the customers, they are very happy with what we have done. In addition to that, they told us they will provide us with additional time, so we can tell them how we have completed the project. The question here is how we did what we have done. From what we have said, it is certain that we still have questions to answer to our customers.

Error Identification and Correction Charts

In order for the error to be corrected, it must be identified; in order for the error to be identified, the communication or the process must be analyzed. We can use the error identification and correction charts listed below to identify errors in our communications.

Error Identification

- 
- Misuse of Instructions
 - Use Inappropriate Instructions
 - Disregard Instructions
 - Misuse of Objects
 - Use Inappropriate Objects
 - Perform Inappropriate Actions
 - Misinformation
 - Misinterpretation
 - Follow Others
 - Miscommunication
 - Exercise of Force

Error Correction

- 
- Good Usage of Instructions
 - Use Appropriate Instructions
 - Regard Instructions
 - Good Usage of Objects
 - Use Appropriate Objects
 - Perform Appropriate Actions
 - Proper Information
 - Good Interpretation
 - Follow Principles
 - Proper Communication
 - Exercise of Kindness

Error in Sentence

- 
- Use of Bad Words
 - Use of Bad Expressions
 - Use of Non Portable Words
 - Use of Non Portable Expressions
 - Use of Words with Multiple Meanings

Correction Consideration

- 
- Use of Good Words
 - Use of Good Expressions
 - Use of Portable Words
 - Use of Portable Expressions
 - Use of Words with Single Meaning

Exercises

For some of us who may have questions about the warning messages, the following exercise can be used as a verification of our understanding of the principle. By having a good understanding of the principle, there should be no problem or ambiguity to verify the warning messages. Also, people who have a good understanding of the principle and who have worked out various exercise from the beginning to the end of the book, should have no problem with the error messages. The understanding of those error messages can be used as a verification to determine whether or not the principle of communication is understood. For some of us who have some difficulty to understand those error messages, turn them off and start working some exercises from the beginning to the end of the principle of communication book.

Since any entity can be used according to us and what we are doing, the exercises are not in order in terms of weights. We can do whatever we think we understand and leave the rest for later. As we make progress learning and understanding the principle, then we can move to do the ones we have left out. While the exercises are given here, they should not be the starting point of learning the principle. To understand the warning messages, it is better to start learning the principle from the understanding the principle of communication book or with the help of an instructor.

1. Show that a person cannot be deleted or erased
2. Verify that a person cannot be copied
3. Show that a person cannot be composed
4. Show that a person cannot be decomposed
5. Verify that a person cannot be grouped with other entity
6. Verify that a person cannot be rotated or flipped
7. Determine that it is not possible to compose a group of people
8. Show that a word cannot be deleted

9. Show that a word cannot be copied
10. Determine that a word cannot be composed and decomposed
11. Verify that an entity cannot be rotated or flipped. You may need to look at the aspect of the entity itself related to rotation. You may also need to look the function of the entity related to rotation as well.
12. Determine that an entity cannot be composed or decompose
13. Show that the function of our communication cannot be deleted
14. Show that the function of our communication cannot be copied
15. Verify that the function of our communication cannot be composed
16. Verify that an application cannot be deleted
17. Show that an application cannot be copied
18. Determine that the result of an application cannot be deleted
19. Show that the result of an application cannot be copied
20. Show that our communication cannot be composed
21. Verify that the communication function cannot be composed
22. Show that the result of our communication cannot be composed and decomposed
23. Verify that it is not possible to copy and delete a timeline
24. Show that it is not possible to compose and decompose a timeline
25. Show that the progress of our application cannot be deleted or copied
26. Verify that the progress of our application cannot be composed and decomposed
27. Show that a node or a callout cannot be deleted or copied

28. Verify that a node and a callout cannot be composed and decomposed
29. Verify that an error cannot be deleted
30. Determine whether or not a problem can be deleted
31. Show that principles cannot be deleted
32. Verify that a compensator cannot be deleted or erased
33. Show that a compensator cannot be copied
34. Show that principles cannot be copied
35. Show that principles cannot be composed and decomposed
36. Determine that the Error Correction Function cannot be deleted or erased
37. Show that the Error Correction Function cannot be copied
38. Verify that the Error Correction Function cannot be composed and decomposed
39. Verify that an action cannot be deleted
40. Determine that an action cannot be copied
41. Verify that a reason cannot be deleted
42. Show that a reason cannot be copied
43. Show that a reason cannot be composed and decomposed
44. Verify with a practical example that a communication holder cannot be deleted and copied
45. Show that a sentence cannot be deleted and copied
46. Show that communication cannot be deleted and copied
47. Verify that information cannot be deleted

48. Show that it is not possible to copy information
49. Show that it is not possible to compose information
50. Determine that information cannot be decomposed
51. Determine the relationship between the word entity, the sentence entity, and the paragraph entity
52. Verify that a word and a sentence cannot be rotated
53. Determine that a reference cannot be deleted.
54. Show that a reference cannot be compose
55. Verify that a reference cannot be decompose
56. What is the difference between a written word and the word itself? Written word means a word on paper, board, postcard, etc.
57. Verify that information cannot be edited. If you want to, you may provide a practical example.
58. Determine the difference between a single entity and a group of entity
59. Show that a reference cannot be edited
60. Show that a reference cannot be copied
61. Verify that the relationship entity cannot be deleted
62. Show that the relationship entity cannot be rotated
63. Determine that the relationship entity cannot be copied
64. Verify whether or not it is possible to combine people with communication function.

65. Determine whether or not it is possible to group people with application. This is the same as saying that, determine whether or not it is possible to group the people with the application entity.
66. Show that question cannot be deleted
67. Verify that a question cannot be copied
68. Show that a question cannot be composed and decomposed
69. Verify that a question cannot be rotated
70. Show that the answer of a question cannot be deleted
71. Verify that the answer of a question cannot be copied
72. Show that the answer of a question cannot be composed and decomposed
73. Verify that the answer of a question cannot be rotated
74. Verify that the answer of a question cannot be edited
75. Show that a question cannot be edited
76. Verify that the match label cannot be deleted
77. Show that the match label cannot be copied
78. Verify that the point to label cannot be rotated
79. Show that the point to label cannot be copied
80. Show that the agreement label cannot be deleted
81. Verify that the agreement label cannot be copied
82. Show that the inclusion label cannot be deleted
83. Verify that the inclusion label cannot be copied

- 84. Verify that the relationship between two entities cannot be deleted
- 85. Show that the relationship between two entities cannot be copied
- 86. Verify that the relationship between two entities cannot be edited
- 87. Verify that an entity cannot be composed. In this case, you verify your understanding about building or deriving an entity.
- 88. What does a project identification mean to you? In this case you verify your understanding about a project identification.

Appendix A: Introduction to Modeling

Modeling is a part of communication. By including modeling in our communication, we take it to a lower level so it can be understood more easily.

An Introduction to Modeling

Introduction

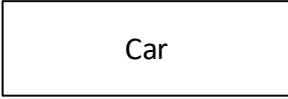
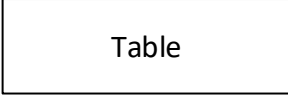
Modeling is a part of communication. By including modeling in our communication, we make it easier to understand. In terms of visual aspect of communication, we identify entities that we are communicating about. It does not matter if the entities are physical or not. Whether an entity is physical or not, modeling enables us to represent that entity by a model. In this case, we can see that, by representing an entity by a model, it makes it easier for us to understand that entity even if we haven't seen that entity or identified before.

In relationship with natural language, incorporating modeling to our communication enables us to take it to a much lower level. The way to look at it, natural language communication is considered to be higher, while the inclusion of modeling of modeling in the communication takes that communication to a lower level so it can be understood much easier. As an example, during communication, a sentence includes one or more words where each word points to an entity or represents multiple entities. If those entities have not been identified by a person before, representing those entities by their models during the communication enables that person to understand that sentence much easier. This is why modeling is important. It takes communication to a level in order to help understand that communication.

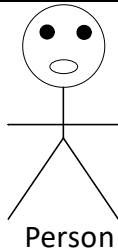
In this chapter, we are going to look at the process of modeling where we can identify entities and represent them by models. For example, using this chapter, we can use models to represent entities that we identify during our communication.

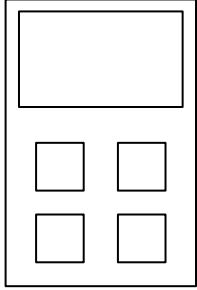
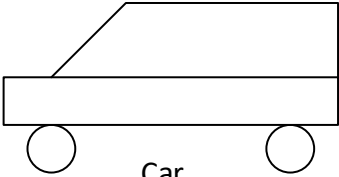
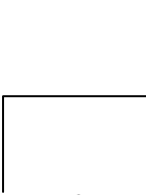
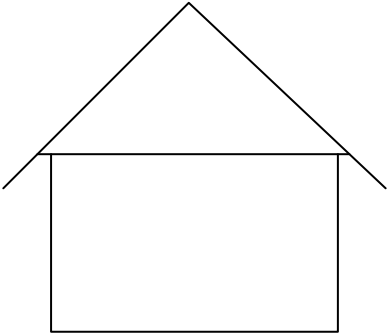
Representation of Entities

In modeling, we represent an entity by a model. In most cases, it is always good to use a box or a rectangle to represent an entity. As an example, let's show multiple entities in the table below and the model representing them.

Entities	Model Represented
Apple	
Telephone	
Car	
Bottle	
Table	
Chair	

As shown from the table above, in modeling to present an entity, we simply draw a box and write the name of that entity in it. It does not matter if that entity is a physical entity or not; we can simply represent the model of that entity by drawing a box with the name of that entity. If possible and necessary, in modeling we can also represent an entity by its own look or its own image. The table below shows an example of entities that are represented by their own images.

Entities	Model Represented
Person	

Telephone	 Telephone
Car	 Car
Chair	 Chair
House	 House

From the table above, we represent the same entities by their natural image. What is important here is that in terms of communication, modeling enables us to identify entities by their natural image. By representing and identifying entities by their natural images, it makes it easier for entities to not be misidentified or misrepresented during communication. If we can represent an entity by its natural model, it makes it difficult for that entity to be misidentified or misrepresented during our communication. Hence, the importance of modeling in our communication. Representing entities by their natural models makes our communication easier to understand and prevents us from misidentifying entities. The way to look at it, during modeling we sketch or represent an entity by providing more information about that entity. It

does not matter if that entity is physically identified or not; we can use modeling to represent that entity as long as we can identify it.

Flow, Connector, Input, Output, and Interface

Introduction

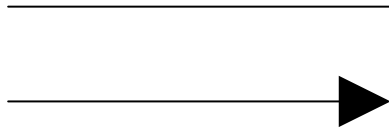
From the previous chapter, we have learned the process of modeling where we represent an entity by a model. By representing entities by their models, we simply identify them. For example, we have identified chair, house, person, telephone and so forth by their own models. In modeling, interfaces and connectors are used to show entities' interactions. For example, two entities are represented where each is identified by its own model. Interfaces or connections can be used to connect both entities to show that they interact or connect to each other.

In modeling, an entity that connects to another entity may send an entity to that entity. We call that flow. Another way to say that is that in modeling, entities may flow from one entity to another entity. For example, if two entities are connected together, another entity may flow from one entity to another entity.

In this chapter, we are to look at the process of modeling where we are going to show entities that are connected to each other as well as entities that flow from one entity to another entity.

Connection of Entities

In modeling, we use lines or lines with arrows to show entities that connect or interface with another entity. Connectors and arrows are represented by the model below.

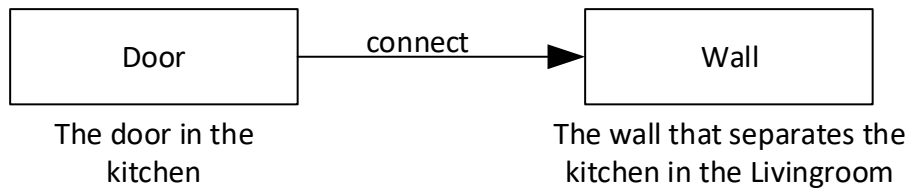


For example, to show that a door connects to a wall, we can use a line or a line with an arrow to show that. The table below provides more information.

Entities	Connections
<div>Door</div>	<div>Door</div> — <div>Wall</div>
<div>Wall</div>	<div>Door</div> → <div>Wall</div>

As we have learned before, it does not matter if the entities physically existed or not. As long as they can be identified, they can be represented by their natural model. In this case, if the entities are interfacing with each other, we can use connectors to show that. Another way to look at it, since communication enables us to identify and present information. We simply identify that information. If that information is identified as two entities that naturally interface, then we can represent that information by a model. In this case, we use connects with the identified entities to present that information.

If we want to, we can use annotation to provide additional information in the model. For example, in the line that connects the door to the wall, we can add annotation to it as well as the door and the wall to show which door and which wall as shown below.



From the model above, we provide additional information about the actual door that is connected to the wall as well as the actual wall that is connected to the door.

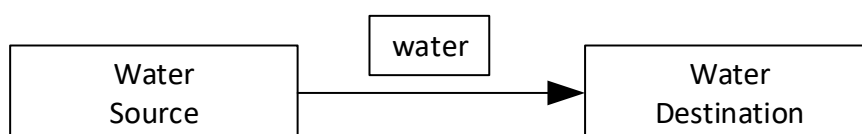
While we use lines and lines with arrows to represent connectors, they are entities themselves. We can also represent them by their own images. For example, a pipe that connects a water source to a destination can be represented by its own image as well as a hinge that connects a door to a wall as shown below.

Entities	Model Represented

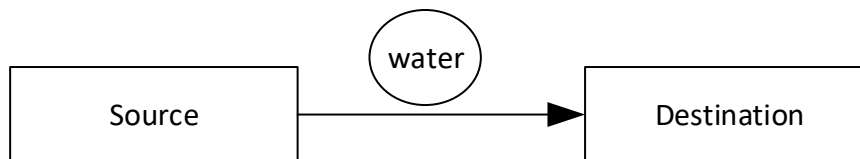
It does not matter if entities are physically identified or not. As long as we can identify two entities that are connected together, then we can use lines or connects to connect the entities to show that they are connected to each other.

Flow of Entities

In modeling, entities may flow from entities to entities. For example, an entity may flow from an entity that connects to another entity. For example, if we represent a water source and a water destination, we can identify water that flow from the source to the destination. In this case, the water source can be a well, a water storage, a water container and so forth, where the water destination can be a faucet, a bucket, and so forth.



From the model below, it shows that an entity which is water is flown from the source to the destination. While we show that in the form above, we can also show that in the form below.



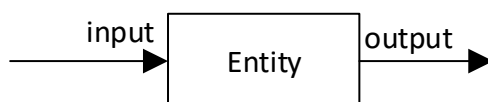
As shown from the model above, in modeling we can use a circle or token to show an entity that flows from another entity. From the model above, we simply use a circle or ellipse to show that water flows from the source entity to the destination entity.

If we want to, we can also use colors to identify the type of entity that flow to another entity. For example, different types of entities can flow from one entity to another entity. In this case, we can use specific color to represent each type of entity so they can be identified more easily.

It does not matter whether or not an entity is physically identified. As long an entity flows into that entity, then we can specify that in the model to show the entity that flows into that entity.

Input and Output

Entity can flow from one end to another end. We call the location where the entity enters, *input* and the location where the entity leaves, *output*. The model below provides an example.



As an example, if we identify a water tank as an entity or a fuel tank for car, where the water goes to the storage is the input and where it leaves is the output.

It does not matter if an entity is physically identified or not. As long as we can identify entities flow in and out that entity, then we can use inputs and outputs in our model to show entities that flow in and out from that entity.

Aspect and Function Representation

Introduction

Modeling is a part of communication; it enables us to take our communication to a lower level so it can be understood more easily. In the previous chapters, we represented entities as we identify them. For example, we represented a car by a model as well as a chair by a model. From the previous chapters, we represented the whole entities we have identified them by their own models. We did not represent the entities we have identified in terms of partition.

In terms of identifying entities, since modeling enables us to lower the level of our communication so it can be understood more easily, it makes sense for us to break down the entities that we identify so they can be understood more easily. Since modeling enables our communication to understand much better, it makes sense for us to model the entities we are communicating about in a way where we can identify parts that make up that entity so it can be understood more easily. In terms of breaking down an entity we are communicating about, when we identify an entity, naturally that entity includes aspects and functions. The aspects of that entity are considered to be parts of that entity so do the functions of that entity. In modeling, it is always good to separate those entities—the aspects and the functions—from the actual entity. Doing so enables us to understand the whole entity much better than otherwise.

In this chapter, we are going to look at aspects of entities and functions of entities. We are going to model both aspects of entities and functions of entities as separate entities. By treating aspects and functions of entities as separate entities, it is possible for us to represent each of them as a separate entity from the model.

Representing Aspects of Entities

In modeling, we simply represent the aspect of an entity as an entity itself. Previously, we represented an entity by a box. We can also use the same box to represent the aspect of an entity. For example, if we identify a water tank that is round, then round is the aspect of that tank and we can identify it by a box as well as shown below.



From the model above, Tank 1 is the actual tank that we identify where Round is an aspect of that tank. In this case, we can see that Round simply provides more information about the tank. What is important here is that the aspect of an entity provides more information about that entity so we can understand it more easily. Having the entity itself without its aspects enables us to know little about it, where the inclusion of the aspect of that entity in the model enables us to know more about that entity, since it provides us with additional information.

In modeling, the aspect of an entity is an entity itself. In modeling, the aspect of an entity is a part of that entity. While we represent the aspect of that entity above as a separate entity, if we want to, we can also use annotation with the tanks to provide such information. It is always good to think that the aspect of an entity is separate from that entity. The aspect of an entity enables us to identify it by distinguishing it from another entity. This is why aspects of entities are important in communication.

In terms of aspects of entities, it does not matter if an entity is physically identified or not, as long as we can identify that entity and the aspect of that entity, then we can show the model of the aspect of that entity. The way to look at it, entities do have aspects. It does not matter if they are physically identified. In our model, we can represent aspects of entities when necessary to provide additional information about them.

Aspects of entities are important to us because they enable us to identify the entities. For example, the aspect of a water tank enables us to identify that water tank—or the type of the water tank. If we can properly identify the aspect of an entity, then we can properly identify that entity.

Representing Functions of Entities

Entities do have functions. The function of an entity is a part of that entity and it is separate from that entity. In modeling, we can represent the function of an entity by a box. As an example, let's represent the function of the water tank above as shown below.



While we use Store Water above as the function of the water tank, we could have also used other verbs like full, empty, and so forth. What is important here is that the function of an entity is separate from that entity. In terms of modeling, by representing the function of that entity as a separate entity, it provides additional information about the actual entity where communication about that entity can be understood more easily.

In terms of aspects and functions of entities, it is important to know that the aspects of an entity exist with that entity, so do the functions of that entity.

Group, Set, List, and Collection Representations

Introduction

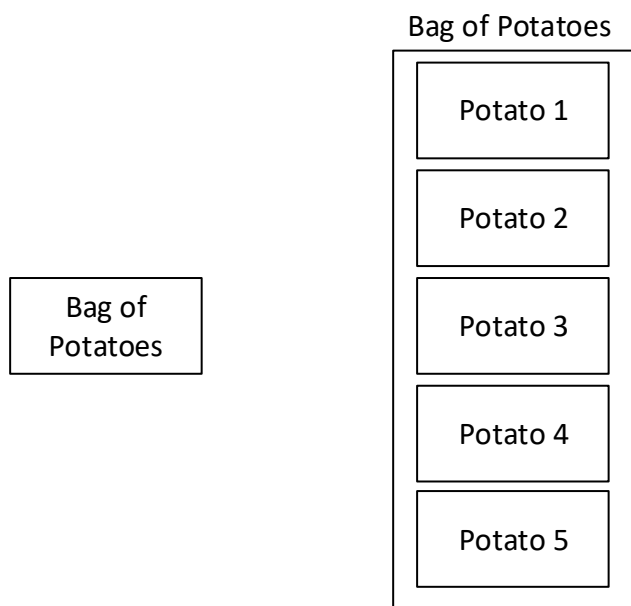
When we identify entities, it is possible for us to identify single and multiple. Naturally, entities are identified as single and multiple. When we identify a single entity, we think of a single entity and when we identify multiple entities, we think of multiple entities. In modeling, we model entities as we identify them. When we identify a single entity, we can represent that entity as a single entity in our model and when we identify multiple entities, we represent them as multiple entities. In terms of multiple entities, we can represent them in our model as lists, sets, groups, or collections.

Previously we identified many entities where we represented them by a model. For example, we identified an actual entity, where we represented it by a model, and the aspect, and function of that entity. Since using modeling in our communication enables it to be understood, it makes sense for us to represent the model of the entities that we identify to help them to be understood in our communication. From the previous chapters, we have shown the model of representing single entities. In this chapter, we are going to show the model of representing multiple entities. The way to look at it, if we identify multiple entities, we can represent them by a model. In this case, by representing multiple entities we are identify by a model, we show that we are talking about multiple items where we provide more information about them.

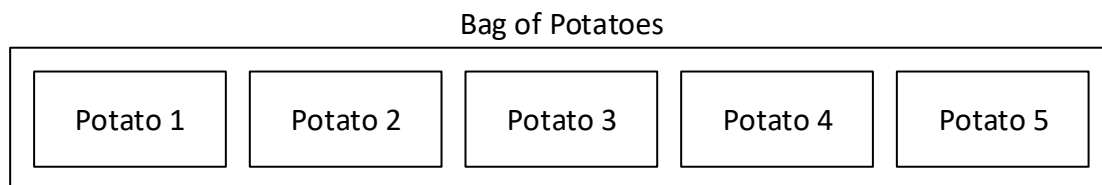
From multiple entities, we can also identify a group of entities. In this case, we can use modeling to identify a group of entities. In this chapter, we are going to look at set of entities, subsets, subgroups, and so forth. We may use the words group, set, list, and collection; it does not matter since all of them have the same meaning. Either word can be used to represent multiple entities. In addition to that, we are going to look at entities and their interactions as well as feedback and adjustment of entities.

Groups, Set, List, and Collection of Entities

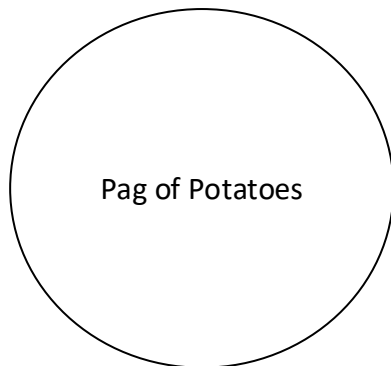
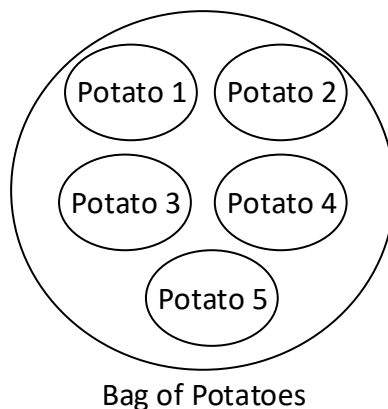
In terms of representing entities, when we identify multiple entities, we can represent them as a collection of entities. For example, a bag of potatoes with 5 potatoes can be identified to the left as a bag of potatoes, where to the right we can identify that bag of potatoes as a collection or a list of potatoes.



As we can see from the model to the right, by representing the model of the bag of potatoes with the including of the potatoes inside the bag, we simply provide additional information about the potatoes which helps with the understanding of the communication about the potatoes. To represent a collection, we simply use a box or rectangle where we include boxes or rectangles inside the box or rectangle. We can call the outer box the main box. We can also represent the same bag to the right in the horizontal for as shown below.



In modeling, we can represent a collection by a circle or ellipse where we can identify items inside that collection as circles or ellipses. For example, the bag of potatoes can be represented in the form below.



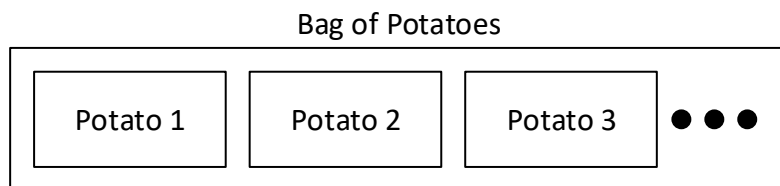
When we represent entities inside an entity, each entity inside the entity is considered to be part of that entity. For example, from the model above, each potato is considered to be part of the bag of potatoes.

Using Continuation in Collection

Since using modeling enables our communication to be more understood, it makes sense to keep things simple. In this case, if possible or necessary, we can use continuation to show more entities that are not listed in the model. For example, if space is an issue, it is better to use continuation to identify more entities that are not shown in the model. Continuation is represented by 3 dots as shown below.

● ● ●

As an example, to show a bag of 5 potatoes, we can use continuation after showing 2 or 3 in the model to show that there are more potatoes in the bag that are not shown in the model.



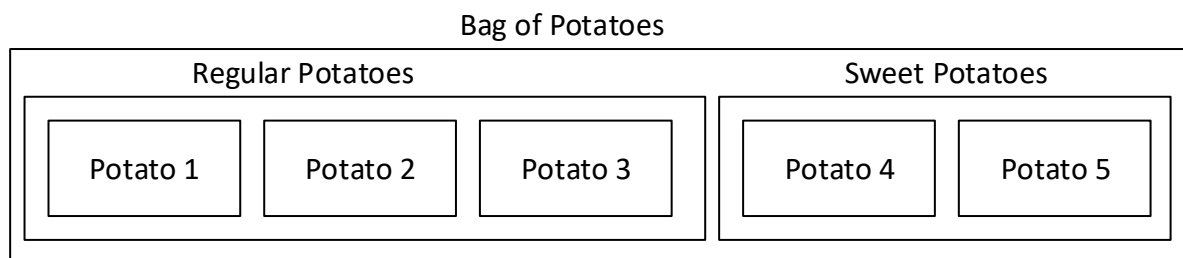
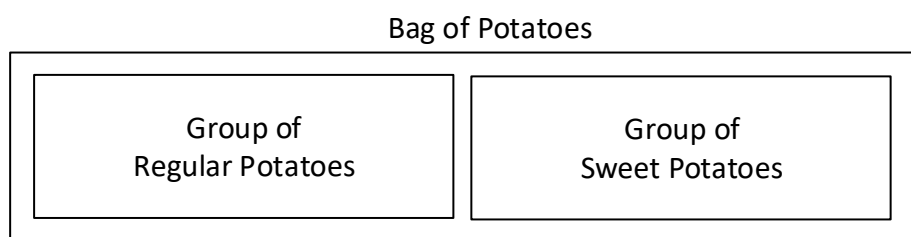
From the model above, while we use continuation with the rectangular representation, it does not matter. We can also use it with the circle representation of the bag of potatoes to show more potatoes that are not in the model.

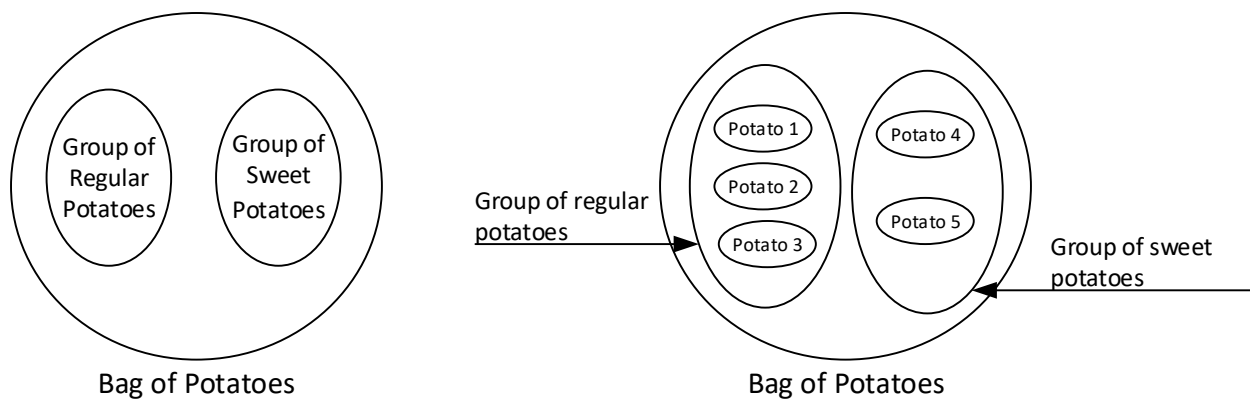
Using Indexing

We can use numbers with entities to identify them in a group or list. For example, in a bag of potatoes, we can identify individual potatoes as Potato 1, Potato 2, and so forth. As shown from the model above, we use index or number to identify individual potatoes in the bag of potatoes.

Grouping of Entities

In terms of modeling, grouping may not have a specific symbol of representation. It may depend on the type of entities. For example, the representation of entities by a box with shadow may mean a group of entities. The representation of multiple persons by “we” may mean a group of persons. We can also circle multiple entities inside a main entity to represent a group of entities. For example, from the bag of 5 potatoes above, if we have 3 regular potatoes and 2 sweet potatoes then we can represent them as a group in the form below. In this case, we can use a rectangle or circle to represent the groups.





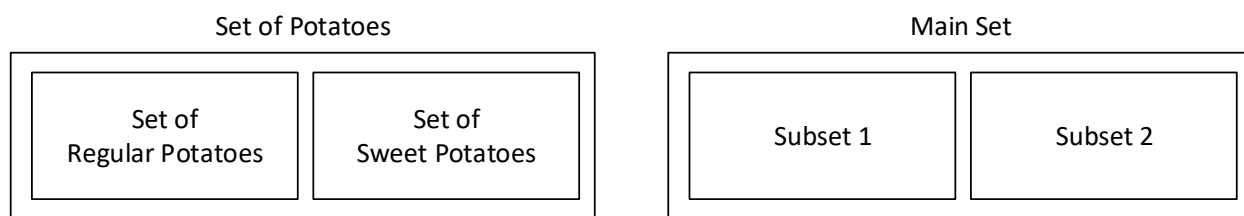
As usual, it does not matter if the entity is physically identified or not. If we identify it as multiple, then we can model it as a group or multiple groups. All that is important is that we model an entity based on our identification of that entity.

Set, Subset, and Main Set

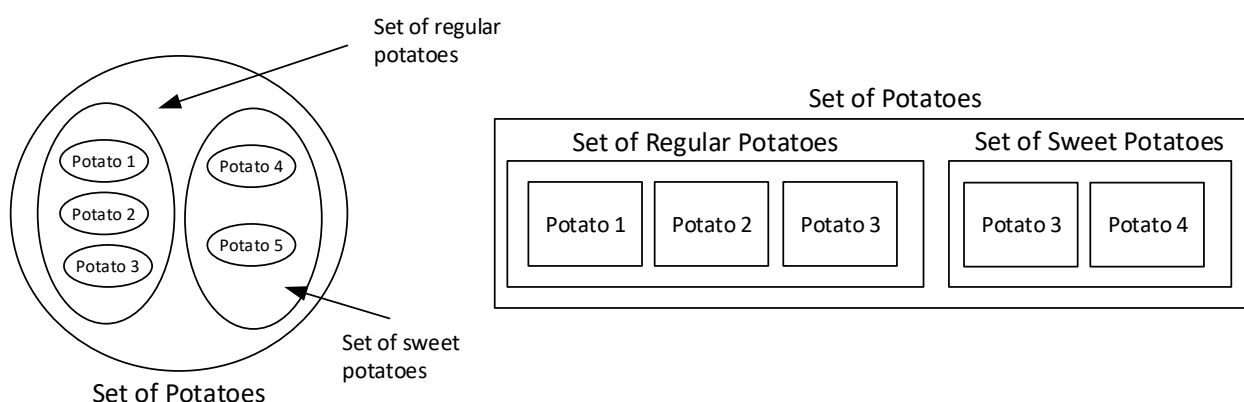
From the example above, we represent a group of potatoes as a set of potatoes. In this case, a group of entities is represented as a set of entities. From the model above, the bag of potatoes itself is viewed as the main group or the main set, where the inner groups are viewed as subsets or subgroups. To better understand that, let's take it like this. Let's assume that in the bag of potatoes above, we have a large bag that includes 2 little bags: one of them is a bag of regular potatoes and the other one is a bag of sweet potatoes. Let's show that by the model below to better understand what we are talking about.



From the model above as we can see, the large bag of potatoes includes two small bag inside where each little bag is viewed as subset of potatoes. As we can see, each subset is identified as two separate type of entities. In terms of main set and subset, we can also represent them in rectangular form.



We can identify individual potatoes inside each set as shown by the model below.



From the model above, we can identify the sets as group of potatoes, list of potatoes, collection of potatoes, and so forth. It does not matter; a set is a group, list, collection, and so forth.

While from the model below we only show two subsets inside the main set, there is no limit in terms of main set and subsets as well as sublevel. For example, if we identify an entity as a main set that has multiple subsets where the subsets also have other subsets, we can also represent them by the model above. What is important here is that an entity can be presented with multiple subsets insider subsets; we can use the same approach to model that entity.

It does not matter whether or not the entity is physically identified; we model that entity based on our identification of that entity. In this case, if we identify that entity as a set, we can model that entity as a set as well as if we identify subsets from that entity, we can also model that entity accordingly.

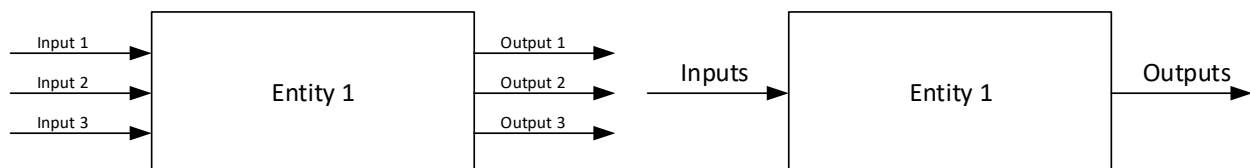
Grouping of Entities and Types

When identifying entities in a set by grouping them, it is important to pay attention to the type of those entities. When grouping entities in a set, it is always good to group them by type. For example, from the set above, we identify 2 types of entities, so we have two groups. If we have

identified other types, we could have had other groups as well. What is important here is that when we group entities, it is important to take their types into consideration. It is always good to group entities that are of the same type as one group.

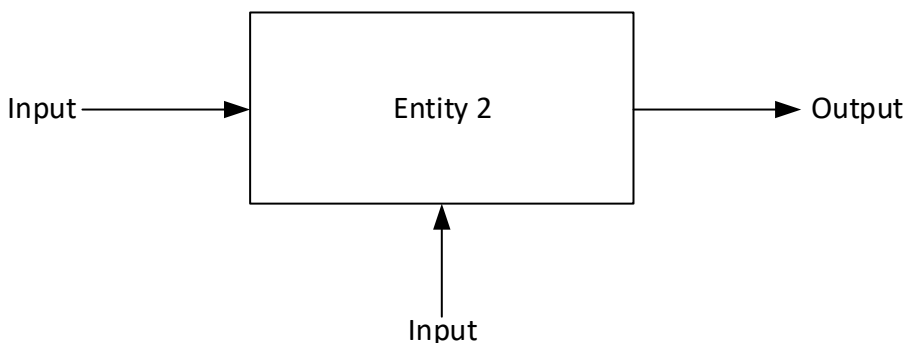
Multiple Inputs and Outputs

Previously we identified a single input and output of an entity. Usually in modeling, multiple inputs are grouped as a single input and multiple outputs are grouped as a single output. If we want to, we can represent multiple inputs and multiple outputs of an entity as shown below.



From the model above, to the left the identified entity has multiple inputs and multiple output. To the right, the inputs and outputs are grouped to show a single input and a single output.

While input of an entity is always shown in the left side, depending on the type of entity we are modeling, input can also be shown on the bottom. For example, if the entity we are modeling includes two types of input, we can put an input at the bottom of the entity to differentiate from the one to the left. As an example, Entity 2 below takes two types of inputs: one in the left and one at the bottom. Both of them are different types.

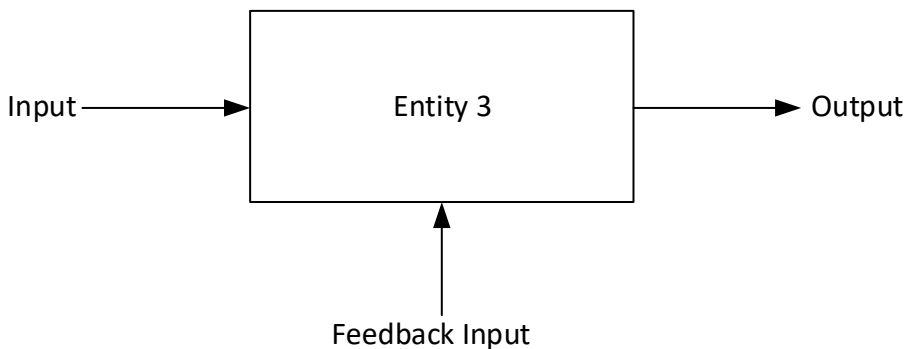


As shown from the model above, the second input at the bottom is identified as an input that is different from the first one. In this case, it does not matter if the entity is physically identified or not. What is important is that the entity takes multiple input and may have multiple outputs.

In modeling, entities that flow from inputs and outputs of an entity are identified as signals. We can call them signals or tokens. We can represent them by different colors to show different types. For example, from the model above, the input to the left can be identified with a token with a different color than the one at the bottom.

Feedback and Adjustment

From the model, we have identified entities that take multiple inputs, where the inputs can be different types and they may appear at different parts of the model. In terms of type of input, if we identify an entity where that entity does have a regular input and another input which can be viewed as a feedback input that enables the adjustment of the output, then we can identify that input as a second type and put it at the bottom of the model. The model below shows an example.



From the model above, Entity 3 is identified as an entity that takes multiple inputs. One of the inputs is viewed as the feedback input. In this case, the feedback input of that entity enables that entity to adjust its output. In modeling, rather than putting that input to the left, since it is different from the regular input, we simply put it at the bottom to show that it is a different type of input that the models take in order to make an adjustment. In this case, the feedback input can be viewed as a valve that enables an adjustment of the output. For example, the model above can be viewed as a water faucet where a valve or adjuster can be viewed as the second input that makes adjustments to the output.

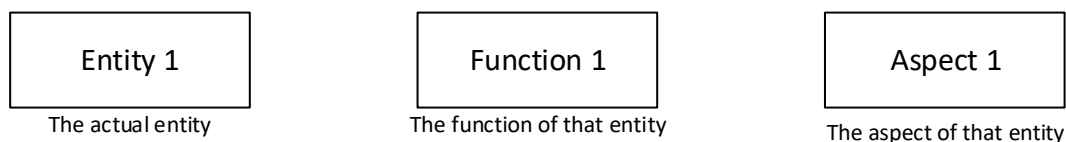
From the model above, Entity 3 can also be viewed as a water heater where the first input is water coming to the heater and the second input is heat or a heating element that enables the heating of the water. In this case, the second input enables the water temperature of the output to change in relationship with the input.

Again, it does not matter if the entity is physically identified or not. As long as the entity takes multiple inputs where it is viewed as feedback that enables the output to change accordingly, then the second input can be identified at the bottom of the model to enable the adjustment.

In terms of entity identification, let's take a water heater as an example, where the first input is water coming to the heater and the second input is heat that enables the heating of the water. In this case, we can separate the first input from the second input and put the second input at the bottom. In relationship with the first input, the output changes accordingly. The second input enables the temperature of the water at the output to be different than the input.

More about Grouping and Collection

Previously we had shown that entities can be identified as a collection or group which we call set. We have identified main sets and subsets which we can call main group and subgroup as well as main list and sub list or main collection and sub collection. We already know that the function of an entity is an entity itself, so is the aspect of that entity. As an example, let's identify an entity as Entity 1 as shown below.

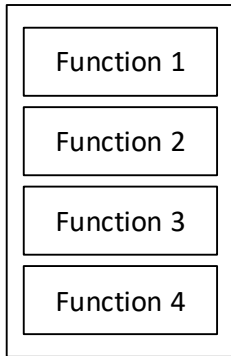


The model above simply shows that an entity that we identify includes two other components that attach to that entity or part of that entity: its function and its aspect. Since the entity itself can be identified as a collection, its function can also be identified as a collection so does its aspect. In terms of function of an entity, if we identify a water tank, that water tank has a function which is *Store Water*. Here, let's represent that entity and its function.

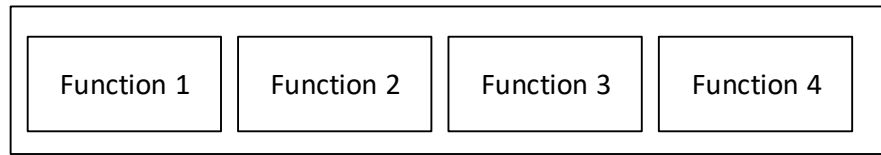


If we can identify multiple functions of an entity, then we can model them by listing them the same way we have modeled when we identified multiple entities. For example, if we identify multiple functions of Entity 1, then we can model them in the form below. If we want to, we can use continuation for more that are not listed in the model.

Entity 1 Function

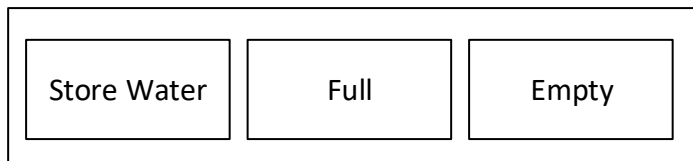


Entity 1 Function

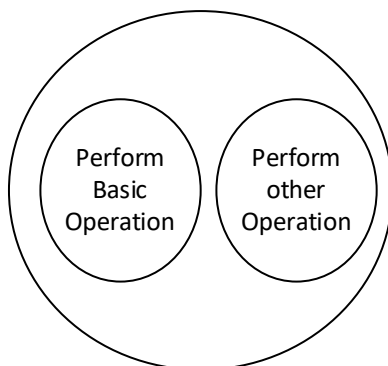


The model above shows that we have identified multiple functions of Entity 1 and we list the functions that we identify. While we represent them in rectangular form, we can also represent them in circular form. In terms of function of entity identification, let's take the water tank as an example. By identifying the water tank, we can identify the following functions: store water, full or is full, empty or is empty. In this case, we can model the functions as a group of functions.

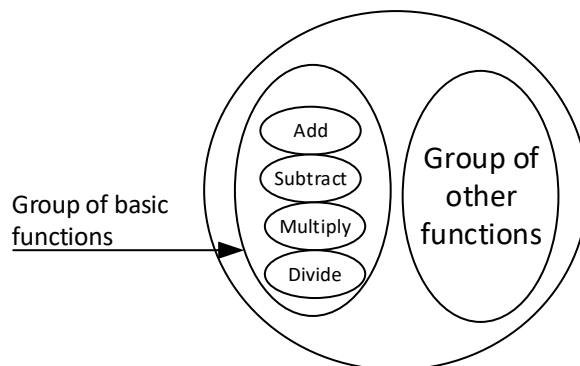
Water Tank's Function



If we identify multiple functions of an entity, we can also group those functions. For example, if we have a calculator that does basic operation and some other functions, we can group the basic operations as one function.



Calculator's Function

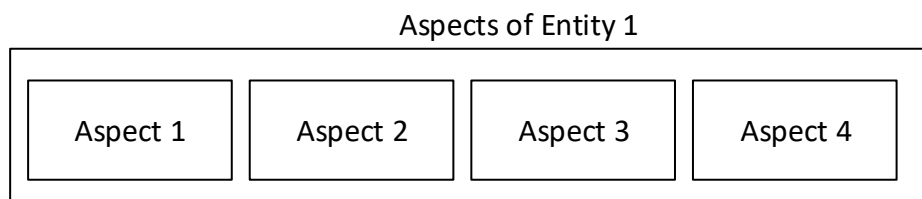


Calculator's Function

From the model above, we identify the basic operation as one group and other operations as another group. While we model that in a circular form, we could have model it in a rectangular form as well.

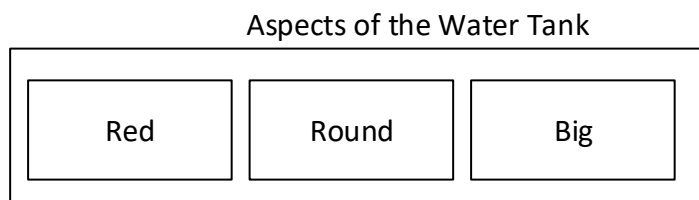
From the model above, each function in the basic operation functions group is considered to be a function of that group. In terms of function identification or entity identification, it is possible for us to identify a function of an entity that is made of multiple functions. In this case, we can call that function main function, where each function of that function is a subfunction. For example, from the water heater model above, store water can be identified as the main function of the water heater where full and empty are considered to be sub functions.

While we identify multiple functions of an entity and model them as a collection of functions, we can do that for aspects of the entity as well. By identifying an entity, we can identify multiple aspects of that entity. By identifying multiple aspects of that entity, we can model them as a collection of aspects. For example, if we identify Entity 1, where that entity has multiple aspects, then we can represent them by a model as shown below.



From the model above, we have an entity that has multiple aspects where we list them in the model to provide more information about that entity. While we show that in rectangular form, if we want to, we can also show that in circular form.

In terms of aspect of entity, let's consider the water tank as an example. We can list multiple aspects of the water tank to provide more information about it. For example, in the water tank that we identified we can see that it is red, round, and big. In this case, we can list those entities as aspects of the water tank as shown below.



From the model above, we can see that aspects of the water tank provide us with more information about the water tank. While we list them in a rectangular form, we could have model that in a circular form if we wanted to. As we did with main functions and sub functions, we can also group aspects of entity as well as identify main aspects and sub aspects of entities.

In terms of modeling functions of entities and aspects of entities as collections, it does not matter if an entity is physically identified or not. As long as we identify multiple functions and aspects of that entity, we can model those functions and aspects as collections and sub collections as well as main functions, main aspects, sub functions, sub aspects and so forth. We can also model functions and aspects as main groups and sub groups as well.

Separation of Entities

Modeling enables us to lower the level of our communication so it can be understood more easily. By including modeling in our communication, we lower the level of that communication where we can provide more information about entities that we identify. Previously, we have identified an entity where we identified both function and aspect of that entity. What we have done is that we separated the entity that we identified into multiple entities where that entity, its aspects, and its functions are considered to be separate from that entity. For example, the water tank we have identified, we have identified the actual tank as an entity, the function of that tank as an entity, and the aspect of that tank as another entity. What is important here is that in modeling, it is important to separate entities that we identify. By separating entities that we identify, it makes it possible for us to provide more information about that entity. For example, by separating the function and the aspect of the water tank from the actual tank, it makes it possible for us to provide more information about tank, so we can make communication about it more understandable.

The reason it is important to separate aspects and functions of an entity from the actual entity, in terms of learning about an entity, it makes it easier for us if we start learning about that entity from its aspects and its function then later learning about the actual entity. By learning or having more information about the aspects and the function of an entity before learning about that entity, it makes it easier to learn about that entity. In other words, in terms of getting information about an entity, if we first get information about functions and aspects of an entity, that provides us with more information about that entity, then later we can understand that entity more easily than otherwise. This is the reason it is important for us to separate entities that we identify.

In terms of separation of entities, it is good to know that the entities do not have to be physically identified. As long we can identify them, it is good to separate them. In other words, an entity does not have to be physically identified or defined. As long we can identify it, its aspects, and its functions, it is good to separate those entities or treat them as separate entities.

Relationship and Mixture Representation

Introduction

When modeling, the entities that we identify can be represented in a relationship form. When that happens, we can represent that entity in the form that we identify it. For example, if we identify that an entity is related to another entity or that entity it is the result of a relationship, we can model that to provide more information.

Previously, we identified functions and aspects of entities and treated them as separate entities. We learned that aspects and functions of entities are important to us since they provide additional information about the entities. In addition to that, it is good to know that relationships of entities are parts of entities and provide additional information about them. For example, if we identify an entity, we identify its function and its aspect, where both functions and aspects of that entity are separate and provide us additional information about that entity. Relationships of those entities are also important to us, and they provide additional information about that entity. Because of that, it is good to model them so we can provide additional information in our communication so it can be more understood. In other words, since modeling enables us to lower the level of our communication to make it more understood, by providing additional information about an entity which includes relationship of that entity with other entities, we further lower our communication to a level so it can be more understandable.

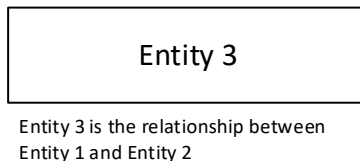
In this chapter, we are going to use relationship and mixture to model entities that we identify. In other words, in this chapter, when we identify entities that are related to each other, we are going to show that by a model so we can understand those entities much better.

Representing Relationship and Mixture

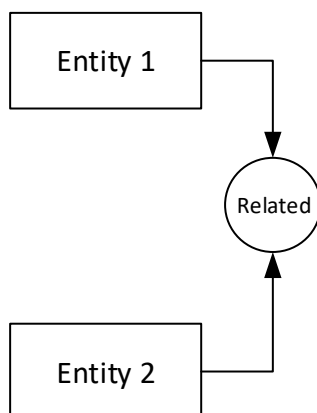
In modeling, we can represent the relationship and mixture entities by the symbols below. It does not matter we can also use the square we have used before to represent relationship and mixture. Usually, we represent the mixture entity by a circle with a cross or X.



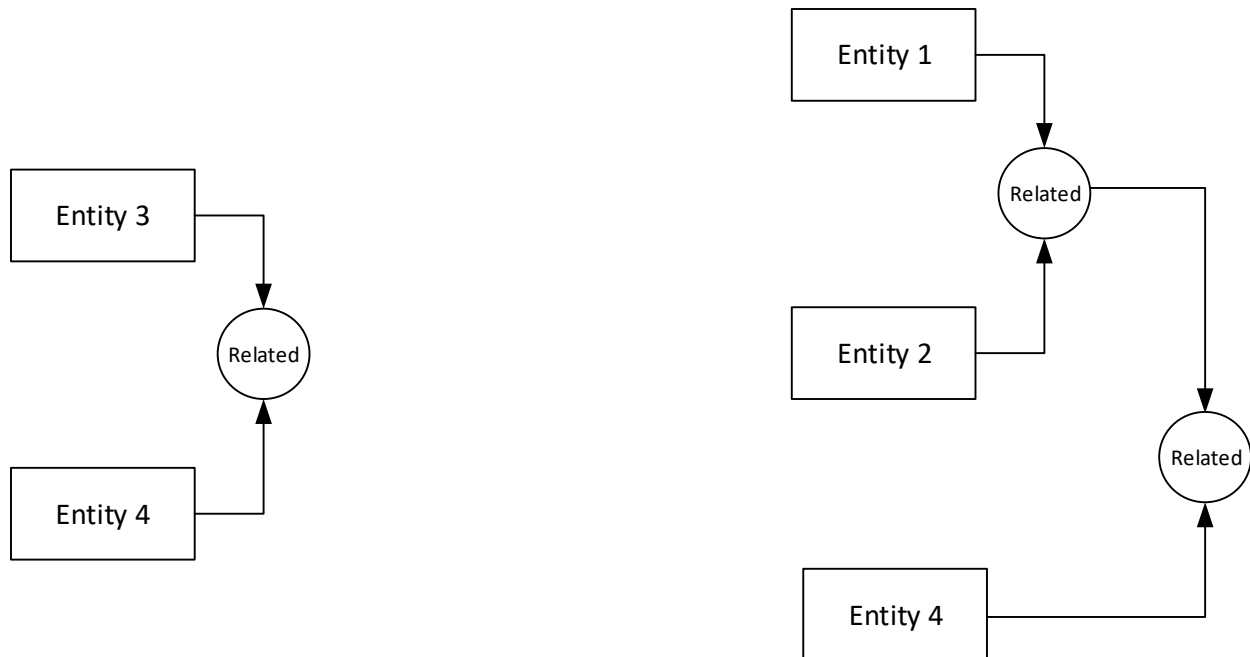
From the model above, we simply use the abbreviation mix for mixture. It is imply a shorter form which takes less space. In modeling, everything we identify from an entity that can provide more information about that entity is an entity itself. If we identify two entities that are related to each other, that relationship itself is an entity and we can show that. For example, if we identify a relationship between Entity 1 and Entity 2, we can represent that relationship as another entity for example Entity 3 as shown below.



If we want to, we can show the whole model of that relationship in the form below. From the model above, we show Entity 3 above as a relationship of Entity 1 and Entity 2. As we can see, the model below provides more information of the relationship than the model above.



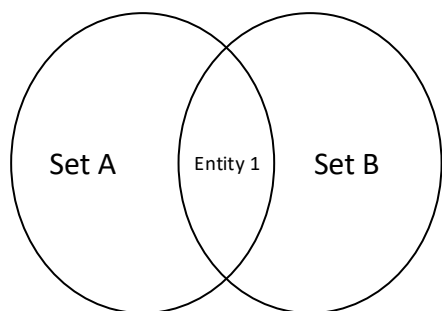
Since the relationship itself is an entity, we can also identify that entity is related to another entity. In this case, we can represent that in the model. For example, if we identify Entity 4 is related to Entity 3, we can represent that by the model below.



As we can see from the model above, both models are the same. The one to the right simply provides us with more information about Entity 3. As we can see, if we have only the model to the left, we know Entity 3 is related to Entity 4, but we don't know enough about Entity 3. In the one to the right, we can see that Entity 3 is the relationship between Entity 1 and Entity 2.

In modeling, the mixture entity is used to show the mixing of two entities. The two entities can be viewed as two input entities that are mixed together to produce an output or a result. For example, the mixture of two entities for cooking, the mixture of two entities to make a shake, and so forth.

When we identify an entity that is produced as a result of a relationship, we can show that as an entity in our model. For collections, groups, and sets, we can identify related entities in the form below. From the model below, it shows both collections are related to each other where Entity 1 is viewed as the entity enables that relationship.



Complexity in Modeling

Introduction

Modeling is a part of communication. By including modeling in our communication, we provide more information of the entity we are communicating about so our communication can be understood easier. From previous chapters, we had shown just that. By modeling an entity, we can provide information about that entity such as: function of that entity and aspect of that entity. During this process, we separated the entities we have identified. For example, we have identified a water tank, where we separate the function and the aspect of that tank. Doing so enabled us to represent those entities separately, so when we communicate about the tank, people can understand us much better.

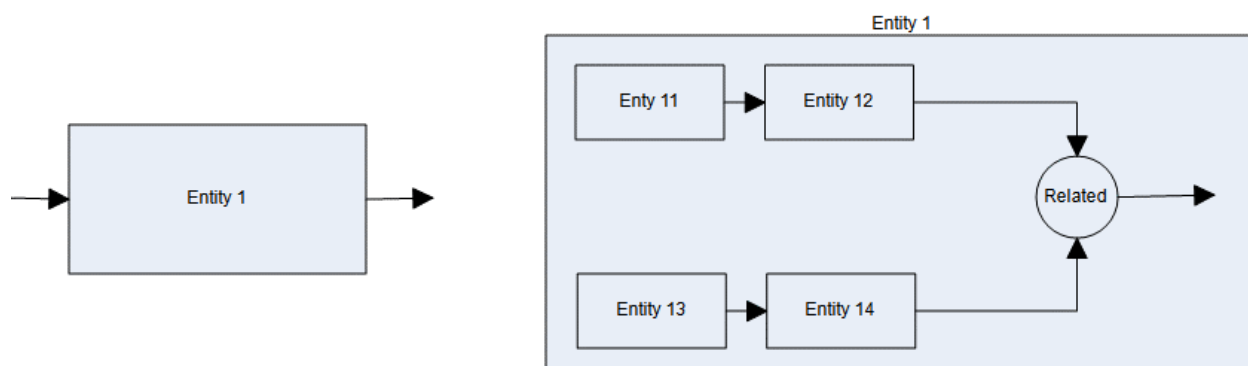
From the previous chapters, we had shown the process of modeling which includes input, output, flow, relationship, mixture, and so forth. Flow of entities in a model enables multiple entities to connect together, where the connections of those entities can represent relationship, as well as other entities such as functions and aspect. Multiple entities can be connected together where signals or tokens can flow from entities to entities. While multiple entities can be connected together to show a model of an entity we are communicating about, it is always good to keep this process as simple as possible. If possible, it is important for us to reduce that model to a form so it can be understood. It is even important to reduce it to the simplest form as possible in order to help others understand the entity we are communicating about.

In terms of reducing a model to a simpler form so it can be understood, it is important as well to understand the purpose of modeling in our communication. As we stated previously, modeling is a part of communication. By including modeling in our communication, it enables us to provide more information about an entity, so that entity can be understood easily. What is important here is that without the model of an entity, communication about it may be a little difficult to understand. By including modeling in that communication, it makes it easier to understand that communication. Because of that, it is important for the model itself not to be confused and make the communication itself difficult to understand. When that happens, the communication itself loses its objective. By understanding that, we can see the importance of reducing complexity in modeling so the communication itself can be understood much easier.

In this chapter, we are going to provide additional information about reducing complexity in modeling so the model itself or the communication can be understood much better. We are going to provide some steps of reducing model to simpler forms so they can be better understood during communication.

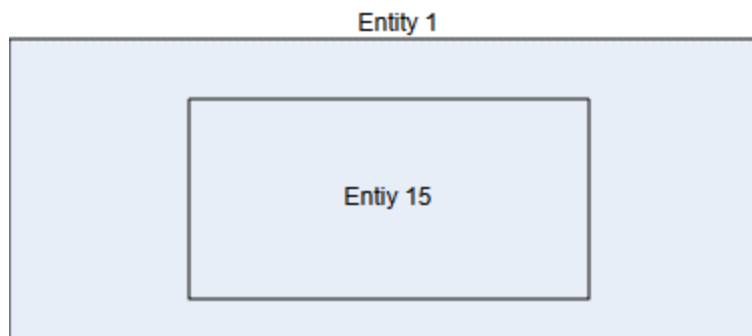
Understanding Complexity in Modeling

When modeling, we simply provide more information about entities that we are communicating about. During this process, we simply identify entities or entities that are relevant to the communication itself or the communication the entity is about. For example, we can represent the model of an entity that is relevant to our communication. This enables us to represent that entity in a form, where it can be understood during that communication. Because of that, the model itself can be limited to that communication. For example, from the model below, to the left we have Entity 1 and to the right we have the same entity.



From the model above, in terms of reducing complexity in relationship to the communication, if the communication is about Entity 1, where the entities inside are not needed to know during the communication, then the model to the left is the only model that is needed. By providing the model to the right, that simply increases complexity in the communication which may enable the communication to be less understood. The way to look at it, during our communication, we provide information about an entity to help us understand our communication. If an entity or entities is/are not needed in the model to help understand that communication, we simply omit it. Doing so enables us to reduce complexity in our communication which is the same as reducing in the model.

From the same model above, let's assume that the relationship inside Entity 1 to the right is identified as an entity and it is Entity 15. During our communication about Entity 1, the relationship itself is not needed by the communication in order to understand that communication. In this case, to reduce complexity in the communication as well in the model, we can replace the model above by the one below.



The way to look at it, as long as the relationship is not needed to help understand that communication, then it can be omitted from the model. Doing so enables us to reduce complexity in the model. In this case, during communication, we provide information about Entity 1 with the inclusion of Entity 15. Since the communication can be understood without saying Entity 14 is a relationship of multiple entities, thus the relationship itself in the model is not needed to help understand that communication. By looking at both models above, replacing the relationship itself by a model simply reduces complexity in the model.

Steps of Reducing Complexity in Modeling

When we model entities, we simply provide more information about entities. During the modeling process, if that information is not needed in the communication, we simply omit it. Doing so enables us to reduce complexity in the model as well as in the communication. Here are some steps to reduce complexity in modeling.

- Reduce the model to the simpler or simplest form whenever it is needed. In this case, if a bigger model is not needed by the communication, use only the simpler model that is needed to help understand that communication.
- In terms of relationship, a simple or simpler relationship is easier to understand than multiple relationships. During modeling, it always good to reduce a relationship to a simpler form so it can be understood. For example, it is easier to understand two entities that are related to each other than multiple entities that are related to each other. During modeling, we can break down relationships two at a time or present relationships in a form where they can be understood much easier.
- Group entities whenever it is necessary. As we have learned before, there are several ways of grouping entities. Grouping entities depend on the entities themselves. Entities

can be grouped by types or by model, and so forth. During modeling, in order to reduce the size of the model to make the communication more understood, it is always good to group them whenever it is necessary. By grouping entities, that enables us to reduce the size of the model as well as the complexity of the communication.

- Break down models whenever it is needed. By breaking down or separating the models, or separating parts of models to represent entities, we further reduce the complexity in the model as well as in the communication. Multiple models can be represented by single model. A single model can be broken down or separated to represent multiple models where we can focus on specific model. This further reduces complexity in the communication as wells as the models.
- In terms of grouping entities, parts of an overall model can be broken down and grouped. Doing so enables the model to be more understood as well as reduces complexity in the model.
- Since modeling is lower than natural language, the inclusion of modeling in our communication enables to lower the level of our communication so it can be understood easier. In this case, whenever it is possible, it is always good for us to represent a model in a flow when it can be read or represented by a sentence. We can call that sentence flown. This process enables us to reduce complexity in the models. In other words, if we can read or represent a model by a sentence, then that makes it easier for us to understand it or understand the communication the model is a bout.

Visual Aspect of Communication in Modeling

Visual aspects of communication enable us to identify entities that we are communicating about. It does not matter if that entity is physically identified or not. As long as we can identify an entity, we can represent it by a model. In relationship with modeling, visual aspects of communication enable us to identify entities where we can model entities that we identify. By modeling entities that we identify, we simply provide more information about those entities in our communication. What is important here is that while modeling enables us to provide more information about entities that we identify, modeling does not allow us to add more information about entities that we identify. The way to look at it, information about an entity that we identify depends on that entity. During modeling, we simply identify and model what we identify. During modeling, we simply identify and model that information. It is not possible or practical to add to that information.

Another way to look at it, since an entity that we identify enables our communication, when we identify an entity, our communication cannot change that entity. When we identify an entity, information about that entity enables us to communicate about it. While we can model that information to help understand our communication, our communication about that entity cannot change that information, since it cannot change that entity. In this case, we can see that while modeling enables us to provide information about entities so our communication can be understood, that information cannot be changed or adjusted by our model.

In terms of modeling entities, as long as we can identify an entity, we can represent it by a model. It does not matter whether or not that entity is physically identified. For example, as we have shown before, a physical entity can be identified by a model, so does a nonphysical entity. An event can also be identified by a model. In this case, if we identify an event, modeling can be used to provide more information about that event. Since modeling cannot change or modify information about an entity, when using modeling to provide information about an event, that information itself depends on that event and cannot be changed or adjusted by our model.

In terms of visual aspect of communication, since modeling enables us to communicate about entities that we identify by providing more information about those entities, modeling itself does not exist without identifying entities. In this case, as long as we can identify an entity, that entity enables our communication and we can provide a model for that entity. As long as we can identify an entity, we can represent that entity by a model. If we don't identify an entity, that entity does not enable our communication and we cannot represent that entity by a model. The way to look at it, the existence of that entity enables the existence of the model of that entity. If that entity does not exist, the model of that entity does not exist. Therefore, our communication about that entity does not exist.

Index

- %
 - % completion, 156
- A**
 - accumulating errors, 142
 - Action Entity*, 40
 - actual communication, 11
 - actual entity, 42
 - Addition, 36
 - Agreement Label, 74
 - Analysis, 25
 - Analysis Logic, 44
 - analysis of entity, 161
 - analysis of entity usage, 161
 - Analysis Window, 30
 - Answer Entity, 67
 - Application Entity, 25
 - Application execution function, 23
 - application is a function of communication, 13
 - application part, 25
 - Application result, 33
 - application status, 156
 - application step, 10
 - Area Entity, 90
 - Area of operation, 95
 - areas of operation, 164
 - Article, 39
 - Aspect of Entity, 95, 96
 - Attach, 76
 - Audio, 44
- B**
 - blue signal entity, 14, 17
 - blue token, 17
 - Book, 44
 - By, 72
 - By Label, 72
- C**
 - Callout, 61
 - Change of application, 101
 - Change of application result, 102
 - Change of communication function, 101
 - Change of Communication Function, 98
 - Change of Communication Function Graphical Form, 101
 - Change of the project, 102
 - Chart, 44
 - Chart Entity, 55
 - cities, 164
 - Cloud, 96
 - Collection of entities, 47
 - collection of words, 46
 - Collection of words, 47
 - Comm Mix, 34
 - communicate together, 163
 - Communication & Application Mixture, 26
 - Communication analysis, 32
 - communication application mixture, 113
 - communication defined as a process of sending and receiving signals, 14
 - communication definition, 10
 - communication domain, 166
 - Communication Domain, 13
 - Communication Element, 43
 - communication entity, 11
 - Communication Entity, 20
 - communication extend, 123
 - communication function, 128, 132, 148
 - Communication Function, 23
 - communication holder, 127
 - Communication Holder, 62
 - communication interaction, 15
 - communication interface, 10, 14
 - Communication Interface, 17
 - communication is natural, 96
 - Communication link, 18
 - Communication Link, 18
 - communication mixture, 119, 123
 - Communication Mixture, 33
 - communication mixture entity, 113
 - communication part, 25
 - Communication path, 18, 19

- communication process, 115
- Communication Process, 29
- Communication purpose, 24
- communication related to what we do, 10
- Communication Related to Work, 77
- Communication Result, 32
- Communication Signals Direction, 78
- communication step, 10
- communication with error, 14
- Comparison, 81
- compensator, 17
- Compensator, 37
- connection of locations, 163
- connection of sites, 163
- content of communication, 14
- continuity, 145
- Continuity, 103
- continuity with application, 146
- continuity with communication function, 145
- continuity with entity, 149
- continuity with function, 148
- continuity with people, 145
- correct errors, 166
- correction identification, 169
- countries, 164
- Curly Brackets, 78

D

- Date line, 51
- dateline, 144
- Dateline, 55
- Definition collection, 48
- Definition Collection, 47
- Definition Entity, 45
- Dependency Label, 73
- detecting error, 166
- diagram, 43
- Dictionary Entity, 46
- Dictionary of definitions, 48
- Dictionary of entity, 47
- Dictionary of word, 47
- Drawing, 43

E

- ECF, 28, 32

- ECF Entity, 44
- employee as communication holder, 127
- employees use entities to execute functions, 161
- empty container, 115
- Empty Container, 64
- Entity, 41
- entity as relationship, 153
- Entity Attach to, 84
- entity produced, 157
- Error Analysis, 25, 32
- Error Communication Gives Rise, 51
- Error Correction, 25, 45
- error correction chart, 169
- Error Correction Chart**, 167, 168
- error correction function, 112
- Error Correction Function, 32
- Error Correction Function (ECF), 25
- error correction mechanism, 32
- Error Entity, 36
- error identification chart, 169
- Error Identification Chart**, 167
- error in communication, 10
- error in sentence, 169
- errors accumulating, 142
- Examples of Modeling**, 112
- extend communication, 123
- external communication holder, 131
- External Communication Holder, 63
- external communications, 132
- extra added to communication, 37

F

- feedback, 146
- Feedback**, 21
- Feedback Entity, 39
- feedback signa, 24
- feedback signal, 14
- feedbacks, 24
- Fixed Entity, 89
- flow chart, 44
- flow of communication, 14
- flow of communication related to work, 53
- flow of signal, 15
- function container, 159
- function of a communication is natural, 96

function of communication, 13
function of entity, 158
Function of what we do, 24
function status, 156
functions execute at separate locations, 162

G

Give Rise Label, 71
Graph, 44
Graph Line and Graph Point, 102
Graphical Axis, 102
green signal entity, 14, 17
green token, 17
Group, 36
group entity produced, 158
group external communication holders, 133
Group of Employees, 103
group of employees use entities, 161
Group of entities, 47
group of errors, 142
group of feedbacks, 143
group of people, 123
Group of People, 103
group of people use entities, 161
Group of words, 47
group people in project, 150
Grouping, 109
Grouping Analysis, 111
Grouping Communication Entity, 110
Grouping Communication Holder, 109
Grouping Entity, 35
Grouping Error and Compensator, 110
grouping of entity, 140
Grouping of Feedback, 111
grouping of problems, 141
Grouping Part of Communication Function, 107
grouping parts of communication function, 116
grouping parts of the application, 116
Grouping Problem, 110
Grouping Questions and Answers, 111
groups of employees at locations, 162
groups of people at locations, 162

H

History, 39

I

identification of that entity is natural, 96
identify a problem, 165
identify a solution, 165
Image, 43
Inclusion, 75
information defined as signals, 14
Information Entity, 38
Information Label, 70
Information Table, 61
Input, 21
Interaction, 76
interaction elements, 15

J

journal, 44

L

Left Signal, 19
List of definitions, 48
list of entities in project/application, 161
list of entities used in the project, 160
Location Entity, 88
Location of operation, 88
locations, 162
locations of operations, 162

M

magazine, 44
main area, 164
main communication function, 117
main entity, 159
main function, 159, 162
Map to, 71
Match, 74
mixture of communication, 29
mobility, 164
Mobility Entity, 88
model, 43
Modeling Examples, 112
Moon, 96

Mother nature, 96
Multiple Entities as an Entity, 84
multiple entities produce a function, 159
multiple relationship, 152

N

Natural Observer, 95
News, 39
newspaper, 44
Node Entity, 58
Node Information, 61
nonphysical entity, 12
Notes Entity, 57
Number Identification, 69

O

operating principle, 155
Operating Principle, 56
Output, 21

P

paragraph, 44
Paragraph, 22
Parent, 96
Parent principles, 25
Parent Principles, 45
Part of an Entity, 85
Part of Application, 105, 106
Part of Application Result, 106
part of the application, 126
parts of communication function, 117
parts of application, 115, 130
parts of entity, 158
parts of function, 158
parts of main communication function, 117
parts of result, 130
people and locations, 162
people use entity to execute functions, 161
people's functions in locations, 162
percent completion, 156
physical entity, 12
Picture, 43
picture of application, 10
picture of the application, 23
Plus, 36

point to arrow, 139
point to label, 139
Point to Label, 70
points of conversation, 137
portrait of our application, 13
principle entity, 155
Principle Line, 56
principle of operation, 155
Principle of Operation, 25
Principles Entity, 24
problem, 165
problem development, 38
Problem Entity, 38
problem identified, 165
problem statement, 165
process of sending and receiving signals, 15
product of application, 157
Progress Entity, 51
progress of application, 119
Project function, 24
Project result, 33
purpose of communication, 10
purpose of the communication, 23

Q

Question Entity, 66

R

Reason Entity, 40
reason of communication, 24
recall communication, 136
recall part of communication, 136
red signal entity, 14, 17
red token, 17
reference, 146
Reference Entity, 65
reference given to, 147
reference with communication, 147
reference with principle, 146
relationship, 152
Relationship, 78
Relationship as an Entity, 86
Relationship Label, 72
relationship with entity, 152
relationship with nodes, 153
relationships one at a time, 154

result of application, 32
Right Signal, 19

S

satisfactory communication, 14
segment of application, 25
sending and receiving signals, 16
sense enables understand communication
 signals, 15
sentence, 44
Sentence, 22
Sentence Analysis, 25, 31
separate application from communication,
 35
separate communication, 10
separate locations, 162
Separation, 49
separation of relationship, 154
Set of definitions, 48
Set of entities, 47
Set of words, 47
signal path, 18
signals entities, 14
Similarity, 81
Site of operation, 88
sites, 162
sites of operations, 162
solution, 165
Solution Entity, 38
solution identified, 165
status of function, 156
Sub Application, 105
sub applications, 130
sub function, 128
Sub Function, 56

sub functions, 117
sub results, 130
Sun, 96

T

table, 44
The Error Correction Function, 28
Time Entity, 56
Time line, 118
Time Line, 50
timeline, 144
transmission of information, 18
two steps, 10

U

Unknown Entity, 68
unsatisfactory communication, 14
Use Label, 77
using the natural observer, 96

V

Video, 43
Visual observer, 96
Voice, 44

W

What I do, 35
What We Do Entity, 34
What you do, 35
Word, 44
Word Entity, 21
word points to entity, 139
Work Entity, 52