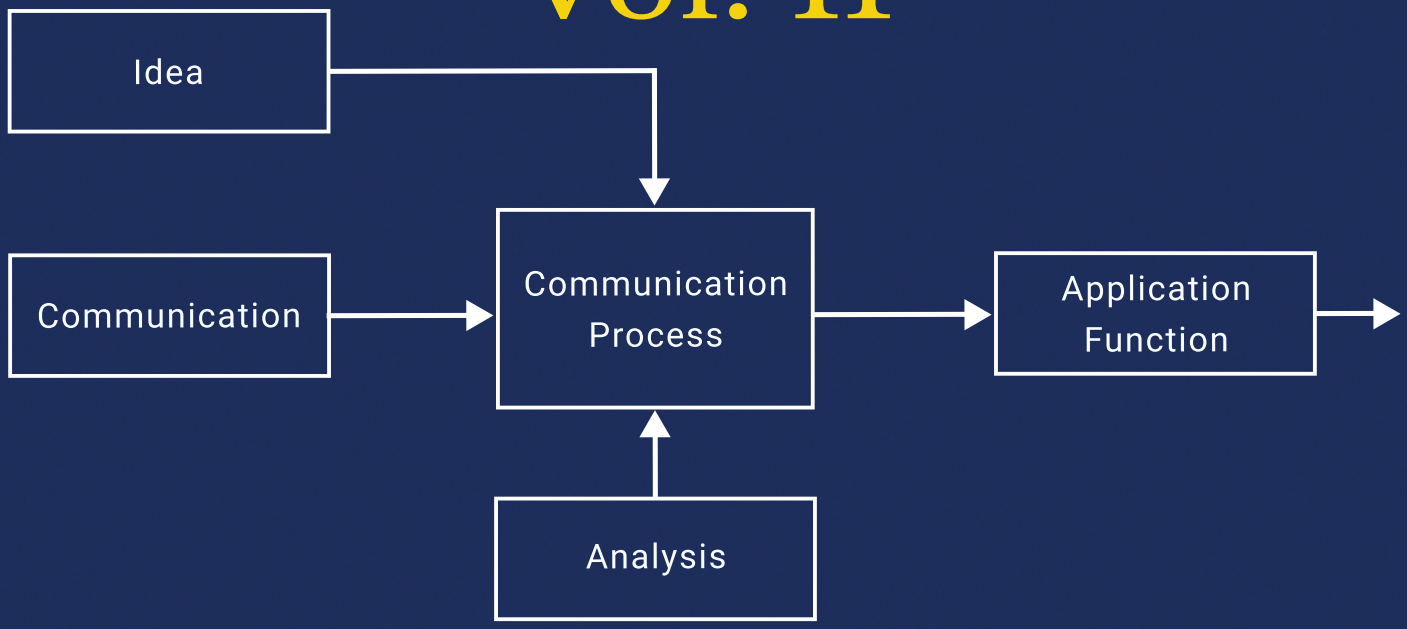
Application Modeling Tutorial & User Manual Communication Domain With Practical Examples Vol. II



The Speak Logic Project

We Promote Better Communication

www.Speaklogic.org

Book #4

The
Speak Logic Project

We Promote Better Communication

Application Modeling Tutorial and User Manual Communication Domain

With

Practical Examples

Vol. II

---

www.speaklogic.org

Copyright 2023 *The Speak Logic Project*

Other Resources

Create and Manage Projects

www.trabau.com

Modeling Software

www.slpsoft.com

Speak Logic Forum

www.speaklogic.org/forum

Download Software www.slpsoft.com

Copyright 2023 *The Speak Logic Project*

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

Copyright 2023 *The Speak Logic Project*

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

Copyright 2023 *The Speak Logic Project*

Table of Contents

Practical Examples 7

Problem Statement 8
Communication in the Application 10
Function of the Application 10
People in the Application 12
People Communication and Timeline 13
Overall Project Communication 14
Function of People in the Application 15
People, Function, and Communication Interactions 17
People Individual Problem Statement 19
Individual Action and Reason 19
Partition of the Application 21
Partition of the Application Communication 22
External Communication in our Application 24
Applying the Principle in our Application 27
Identifying the Principle in our Communication 29
Analysis of Communication 31
Principle Verification 35
Identifying Error in our Application 42
Correction of Error in our Application 46
Application Execution Location 49
Understanding the Correction Process 50
Solution 53
Understanding Problem and Solution 53
Understanding External Function 56
Understanding Function Dependency 59
Understanding our Functions Related to Locations 61
Understanding our Function in our Application 62
Conclusion and Application Verification 63

Understanding Complexity in the Communication Domain 67

Introduction of Complexity in our Application 71
Error in Communication in Relationship with Complexity 72

Copyright 2023 *The Speak Logic Project*

Error Correction Reducing Complexity 73
Complexity and Idea of Application 74
Complexity and People in the Application and their Functions 75
External Communication Related to Complexity 76
Application Performance Related to Complexity 81
Communication Relationship and Application Idea 82
Operating Principle with Reducing Complexity 83
Functions Discontinuation 85

Understanding Resource Management in the Communication Domain 87

Complexity and Resource Management 88
Visual Aspect of Communication and Resource management 91
Problem Identification and Complexity 92
Steps of Reducing Complexity with Resource management 92

Understanding our Parent's Vocabulary 94

Understanding our Parent Dictionary 95
Reduction of Words in our Dictionary 96
Derived Words and Derived Entities 130
Why a Bigger Dictionary is not Needed 135
Reducing Dictionary Size Related to Time 139

Understanding Complexity and our Vocabulary 143

Complexity in Size of our Dictionary 144
Summary 148

Understanding Stability and our Vocabulary 153

Reducing the Size of our Dictionary and Stability 157
Conclusion 157

Index 158

 Copyright 2023 *The Speak Logic Project*

Practical Examples

Introduction

Given that a lot of times we do things by communicating with each other, we can see that the things that we do are driven by our communications. If we break down what we do in relationship with our communications, we can identify multiple components in our applications and how each component in our application is related a specific segments or sections of our communications.

As we have learned from the previous sections, a lot of components of our applications in relationship to our communications have been identified. By using those components, we can model our applications where we can identify specific components and their interactions and how the results of our applications are affected in relationship of those components. In other words, by identifying those components by modeling our applications in relationship to our communications, we can determine the results of our applications based on the value of those components.

To help us understand better the process of modeling our applications, this chapter provides some practical examples about using the modeling process in an actual application. Since we use communication in almost everything that we do, it makes sense to use a practical example everybody can understand. In this case, a practical example of an application does not limit our understanding of the modeling process. Thus, it is good to use an example everybody can understand.

Copyright 2023 *The Speak Logic Project*

The Lemonade Application

As a practical example, let's take a lemonade application for instance. Here to start, we define the problem statement. In the problem statement, we identify the problem we try to solve.

Problem Statement

I feel so hot, I need to make a lemonade to refresh myself. Let's identify key elements in the problem statement and rewrite the problem statement below to identify the key elements.

I feel so hot, I need to make a lemonade to refresh myself

Above, we identify the key elements of our problem statement, let's put them on the table below.

Problem Statement Entities Identified
Feel hot
Make lemonade
Refresh myself (or simply refresh or to be refreshed)

Now, let's use another table where we can identify the elements of the problem statement above and equalize them to the entities that make up a problem statement which are action and reason.

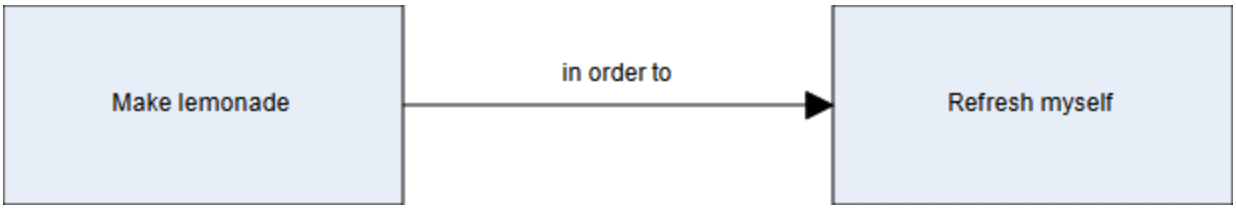
Problem Statement Elements	**Problem Statement Entities**
Make lemonade	Action
To refresh	Reason

Given that, the *make the lemonade* takes the *feel so hot* element into consideration, we did not include it. If we want to, we can include it by showing in the table below.

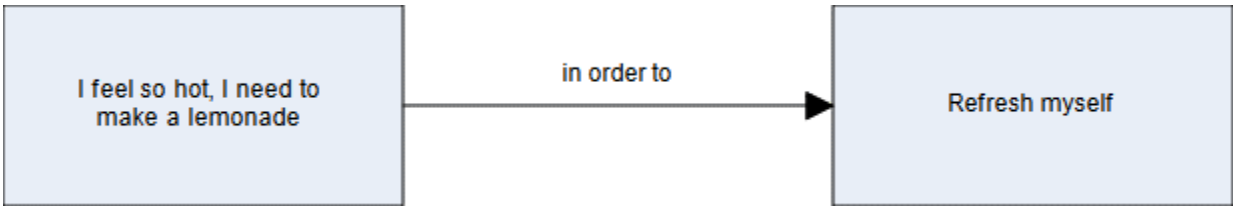
Problem Statement Elements	**Problem Statement Entities**
Feel so hot, need to *make a lemonade*	Action
To *refresh myself*	Reason

Now, let's use the modeling entities to model our problem statement

 Copyright 2023 *The Speak Logic Project*



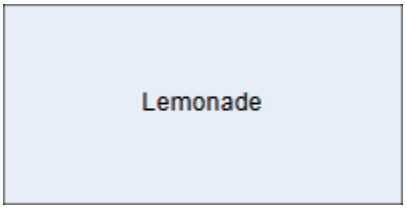
The same problem statement above can also be represented in this form



While it is not necessary, if we want to we can identify the actual problem we try to solve by using the problem entity. Let's use the problem entity below to show the actual problem. Both of them are the same, *feel so hot or feel hot* is the actual problem that's need to be solved.

Now in order to solve the problem, we identify the entities that we need to make the lemonade. In order to make the lemonade, we need *lemon, sugar, water*, and ice. To solve this problem, I look at in the fridge, I don't have lemon, but I have water and ice. I look at in the cabinet and I also have no sugar. In this case, I am going to ask my next-door neighbors for lemon and sugar. I check with Ms. Smith, and she told me she can give me two lemons. I also call Peter my other next-door neighbor and he told me he can get me a cup of sugar.

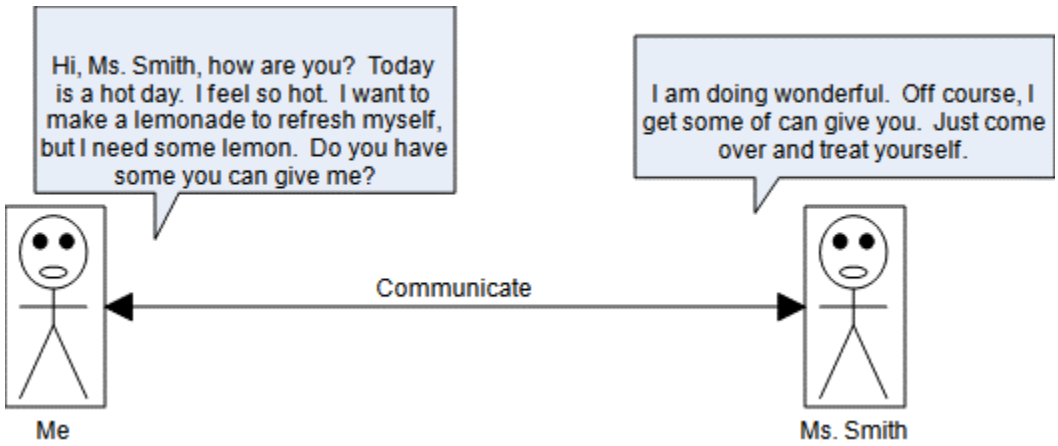
Before we do so, let's identify the application.



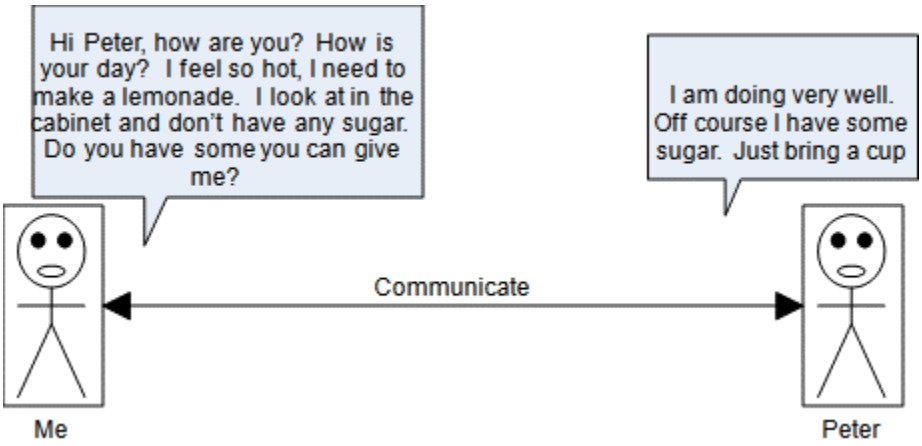
 Copyright 2023 *The Speak Logic Project*

Communication in the Application

Now, let's look at the communication between me and Ms. Smith as well and the communication between me and Peter. Below is the first communication between me and Ms. Smith. I communicate to Ms. Smith about the problem and what I need to solve it, she also communicates with me to give me what I need to help me solve that problem.



Here let's look at the communication between me and Peter about the problem that I need to solve. I communicate with Peter about the problem, where Peter communicates with me and how he can help me to solve the problem.



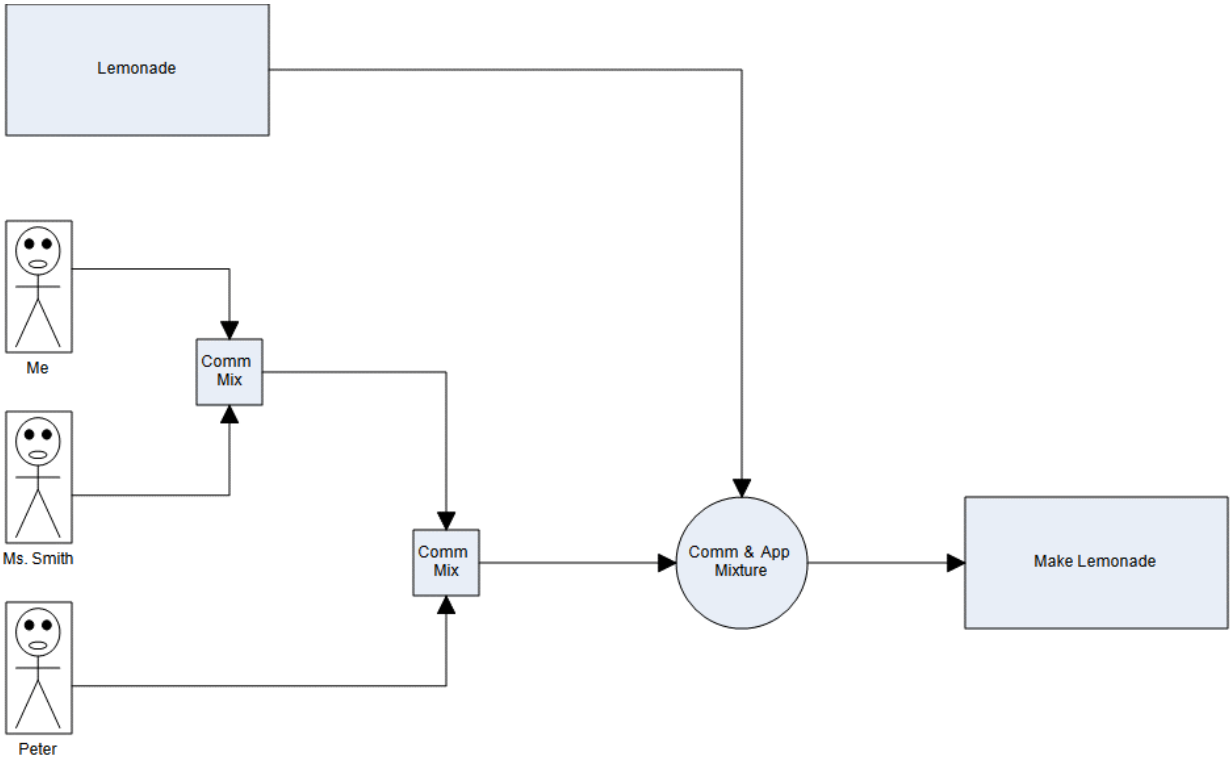
Function of the Application

Here, let's model the lemonade application with Me, Ms. Smith, and Peter as well as the identified communications. As we can see from the communications above, we can identify the

 Copyright 2023 *The Speak Logic Project*

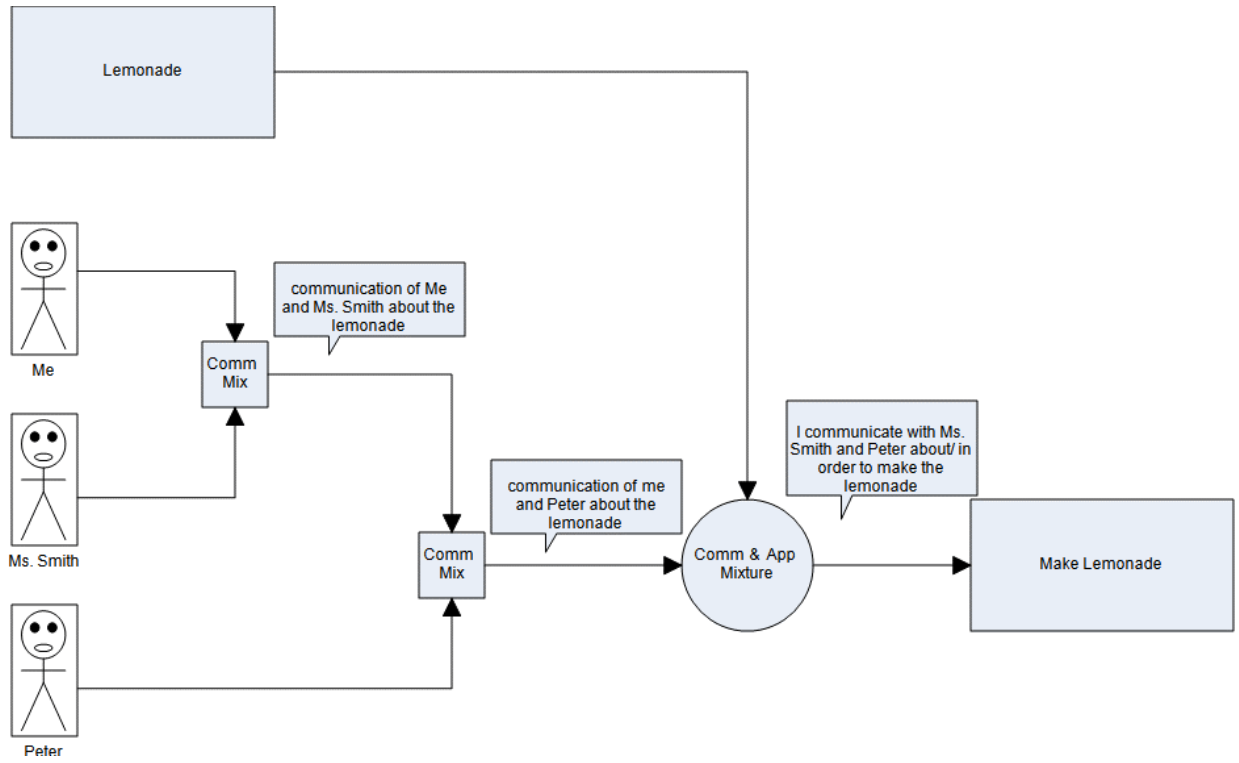
function of the communication which is make the lemonade from the actual communication. Here we can put it below.

Below we show the communication flow of making the lemonade where I communicate with Ms. Smith about making the lemonade and I also communicate with Peter about making the lemonade. The result is the function of the communication which is make the lemonade.



Below, let's show a model with more information about the flow of the communication.

Copyright 2023 *The Speak Logic Project*



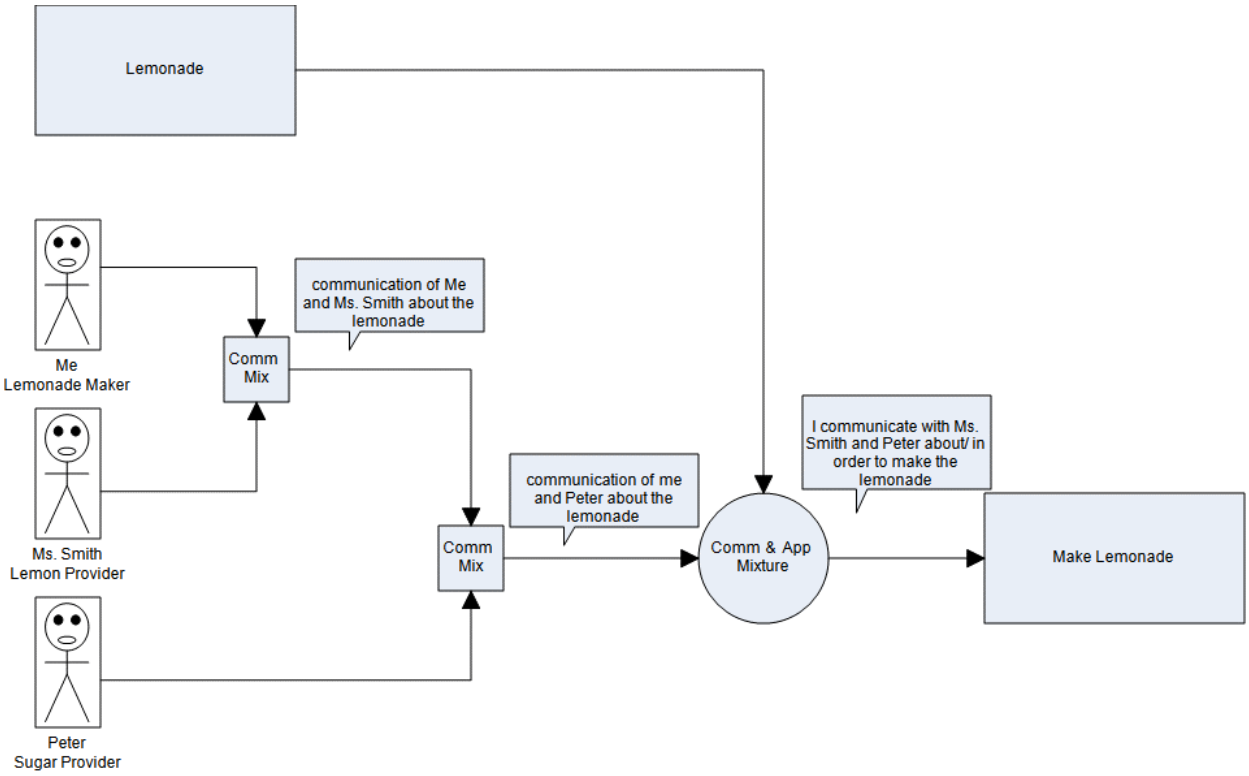
People in the Application

From the information above, we can identify the people that make up the lemonade application as well as their functions in the application. Let's represent them with more information in the table below.

People	Function
Me	Make the lemonade
Ms. Smith	Provides lemons to make the lemonade
Peter	Provides sugar to make the lemonade

The model below shows each person and his/her title in the application. If we want to, we can omit the title and simply put function of the person below the name of the person. It does not make any difference.

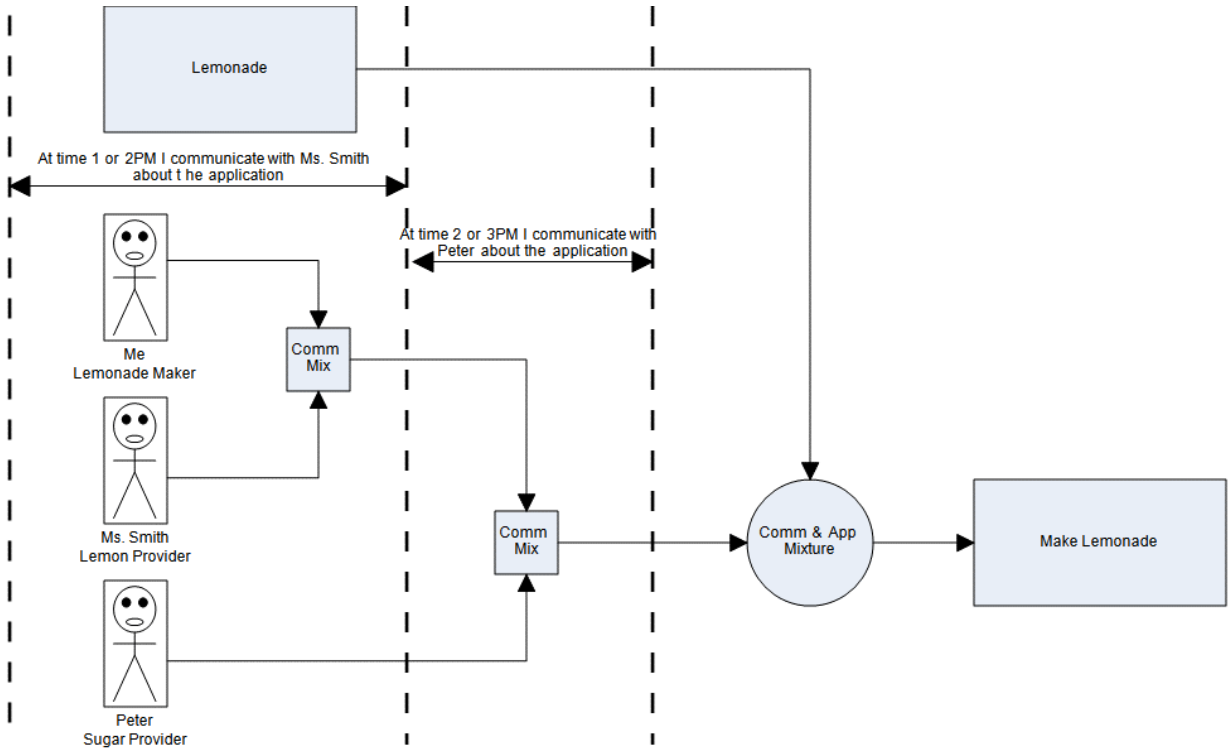
Copyright 2023 *The Speak Logic Project*



People Communication and Timeline

In term of time, let's look at the flow of the communication in relationship with the application. The model below shows an example. At 2 p.m. for instance, I communicate with Ms. Smith about the application. Where at 3 p.m. I communicate with Peter about the same application. As we can see below, the lemonade application does not change at 2 p.m. when I communicate with MS. Smith. It also remains the same at 3 p.m. when I communicate with Peter.

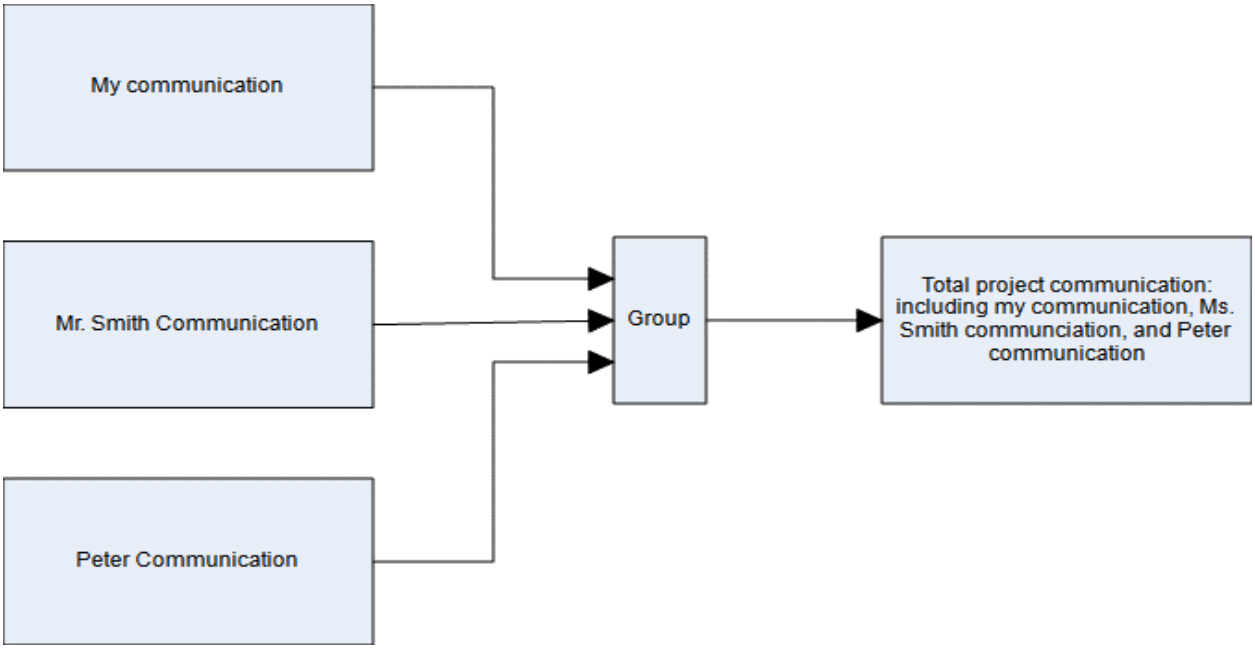
Copyright 2023 *The Speak Logic Project*



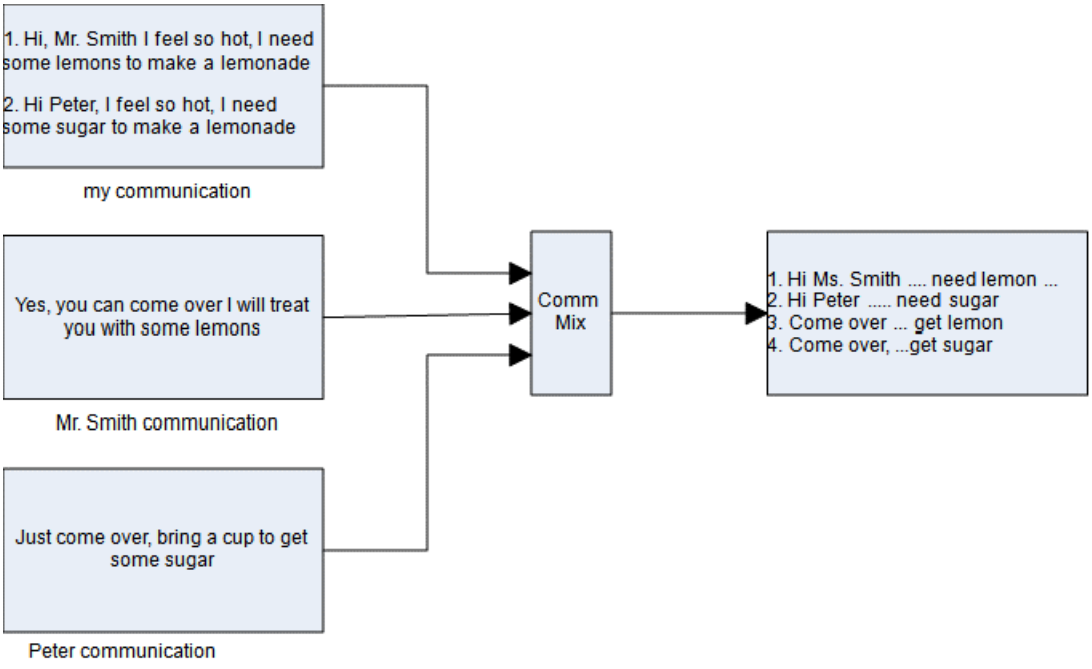
Overall Project Communication

As a separate entity, let's identify each communication and group them together to show a main communication. As shown below, I first communicate with Ms. Smith about the application. This is viewed as the first communication. Second, I communicate with Peter about the application, this is viewed as the second communication. Another way to look at it, is that, my communication, plus Ms. Smith's communication, plus Peter's communication are viewed as the total communication in the application. In this case we can have them as follow.

Copyright 2023 *The Speak Logic Project*



While the model above shows the grouping entity to show all the communications, we can also use the communication mixture entity. Let's show the overall communication of the project with the communication mixture entity.

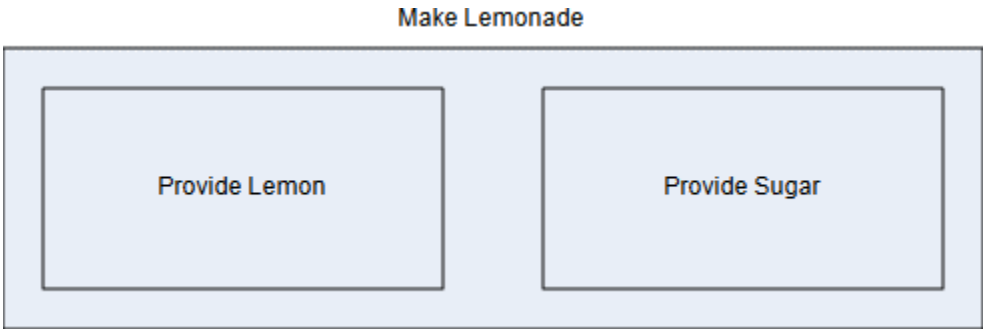


Function of People in the Application

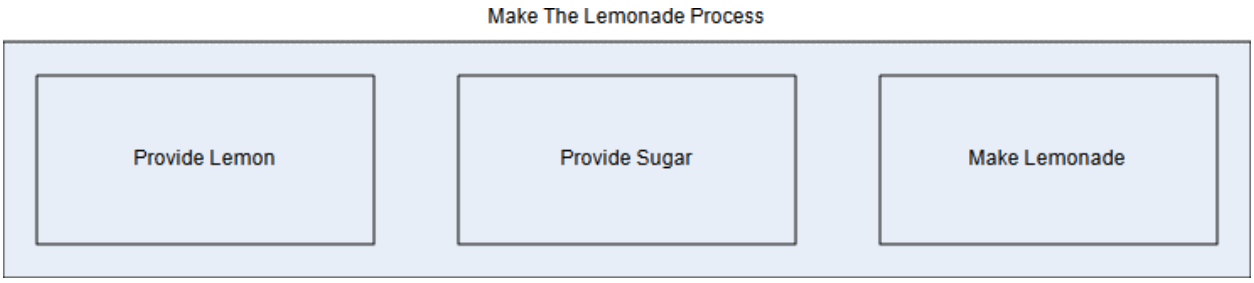
 Copyright 2023 *The Speak Logic Project*

We previously presented a model to show the flow of the communication that enables the lemonade application. From that model, we have identified each person as well as the title of each person. I was identified as the lemonade maker, where Ms. Smith was identified as the lemon provider, and Peter was identified as the sugar provider. Here, let's identify the function of each person as a separate entity.

Above we have shown each function, we can represent provide lemon and provide sugar as sub functions of the make lemonade function as shown below.

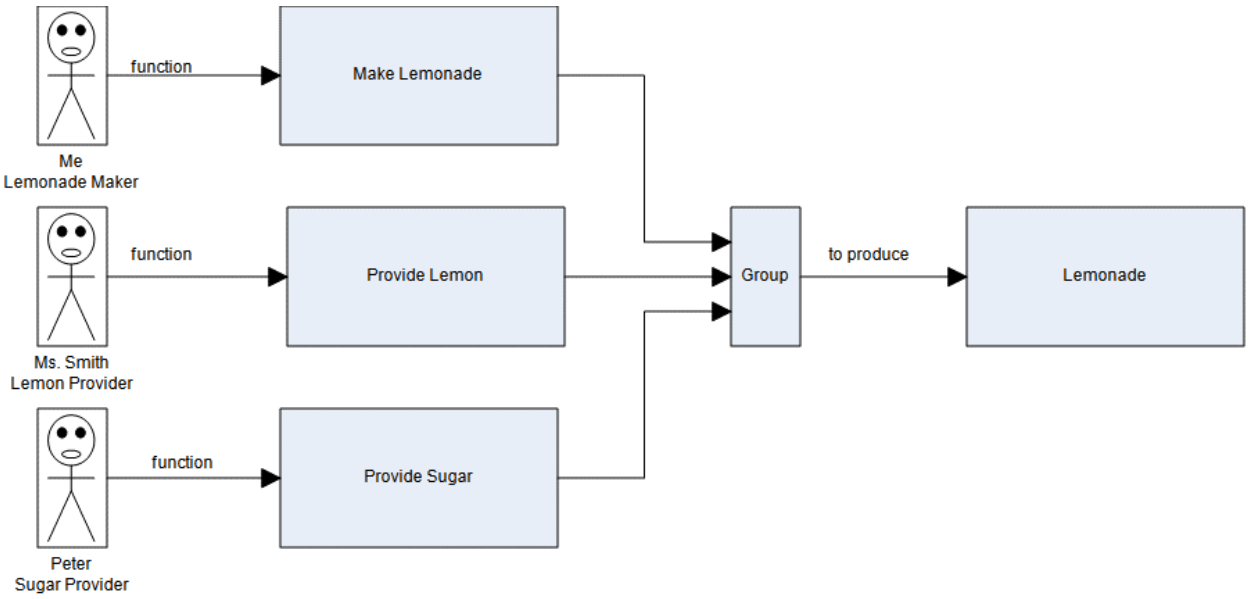


If we want to, we can show the process of making the lemonade as a function, where inside the that function we can have all the functions as shown here.

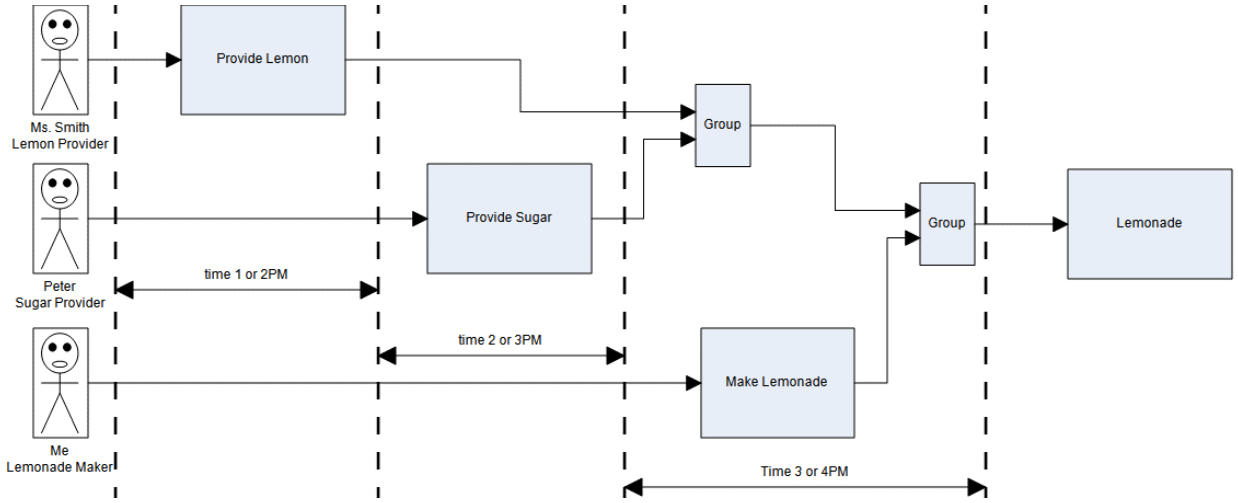


Here, we can identify the function of each person in the application with the application and how they group together to produce the lemonade, which is the result of the functions.

Copyright 2023 *The Speak Logic Project*



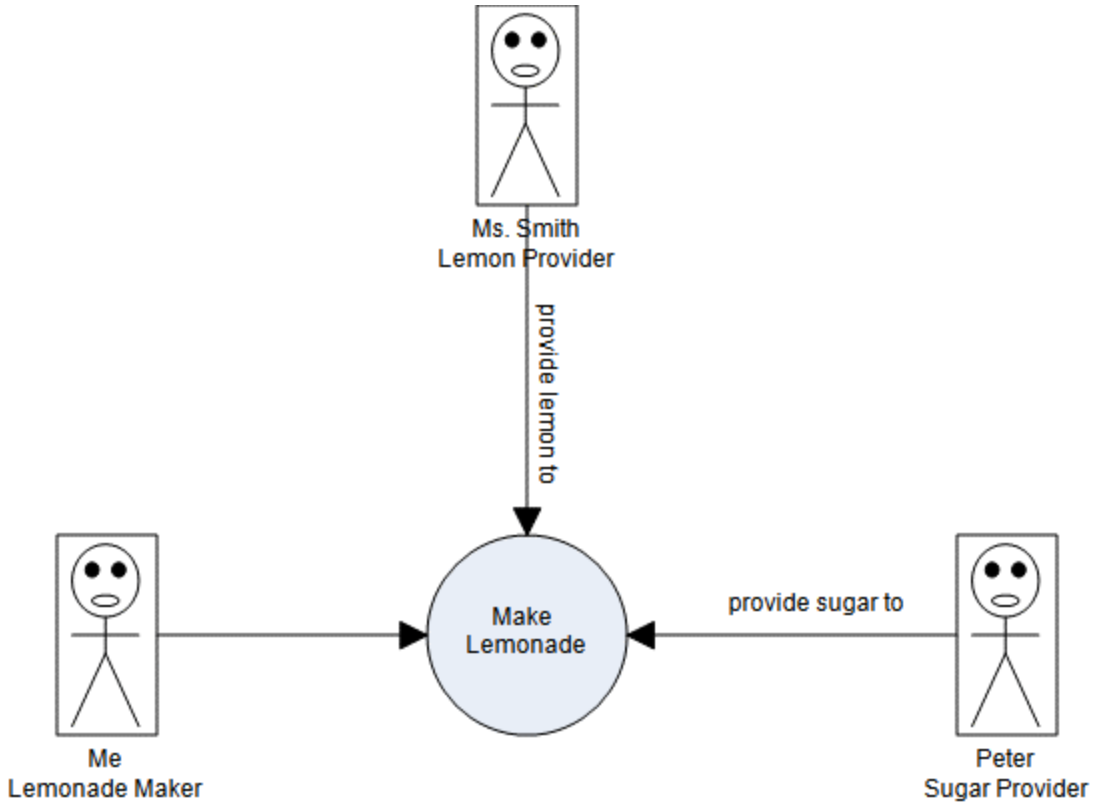
If we want to, we can show time line between the process and use another group entity. The model below shows just that



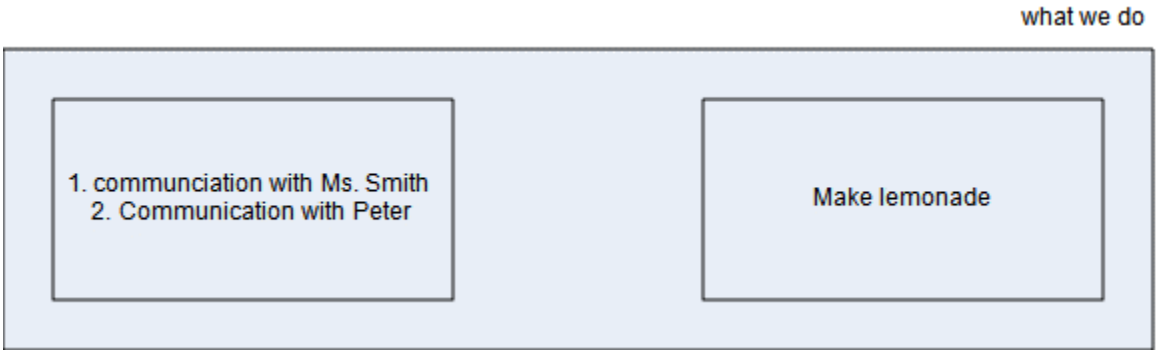
People, Function, and Communication Interactions

As we can see above, we have three people in the application, we can show the model on how they work together to enable the application. Below shows a contribution of each person in the application.

Copyright 2023 *The Speak Logic Project*

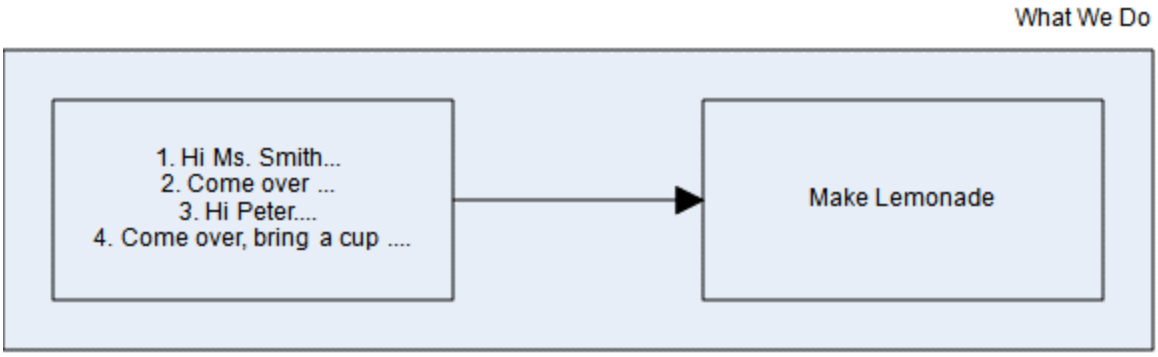


Now, let's use the what we do entity to show what we do in the application. The what we do entity includes the communication entity and what we actually do.



Now, let's show the actual communication which includes my communication with Ms. Smith and my communication with Peter. Below, we identify each communication that takes place as well as the actual application.

 Copyright 2023 *The Speak Logic Project*



People Individual Problem Statement

We have already shown the problem statement for the project. Let's identify the problem statement again *I feel so hot, I need to make a lemonade to refresh myself*. This problem statement is viewed as the overall problem statement for the application or project. Given that we have three people that enable the solving of the problem or that enable the lemonade to be produced to solve the problem, each of them does have a problem statement. Here let's identify the problem statement for each person. Before we identify the problem statement for each person, let's identify the problem each person tries to solve.

Person Identified	Problem Statement/Problem Identified
Me	I feel hot, I need to make a lemonade to refresh myself. Problem identified: *feel hot*, action: *make lemonade*, reason: *refresh myself*.
Ms. Smith	You feel hot, you need to make a lemonade, you don't have lemon to make the lemonade. Problem identified: *feel hot*, action: *provide lemon*, reason: *to make lemonade* (to refresh yourself)
Peter	You feel hot, you need to make a lemonade, you don't have sugar to make the lemonade. Problem identified: *feel hot*, action: *provide sugar*, reason: *to make lemonade* (to refresh yourself).

As we can see above, the problem statement for each person (Ms. Smith and Peter) looks like the main problem statement. That makes sense, since the main problem statement identified the problem to be solved so does the reason to solve that problem, the problem statement for each person must also look similar to the main problem statement to include the same reason as well.

Individual Action and Reason

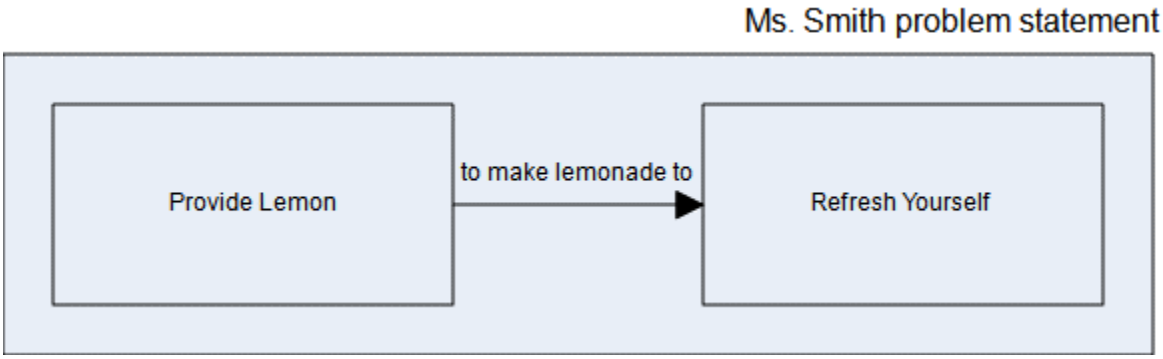
 Copyright 2023 *The Speak Logic Project*

Here in this table, let's identify the action and reason for each problem statement again by using the table below.

Person Identified	Action	Reason
Me	Make lemonade	Refresh myself
Ms. Smith	Provide lemon	*make lemonade to* Refresh yourself
Peter	Provide sugar	*make lemonade to* Refresh yourself

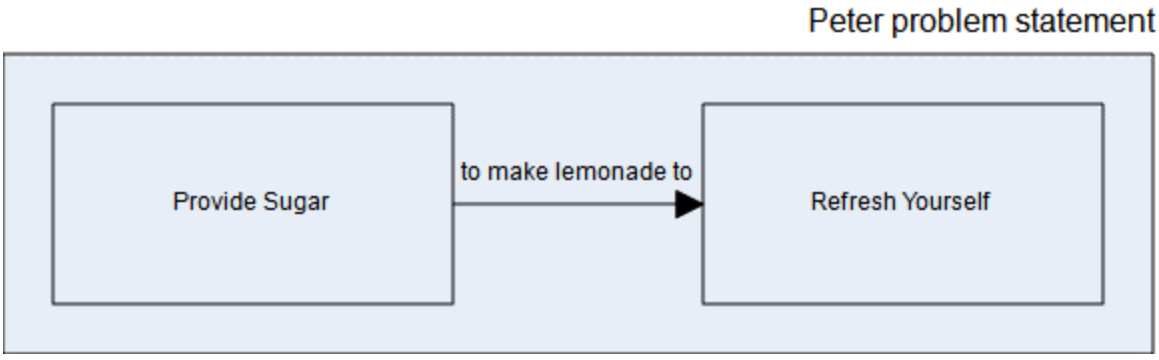
As we can see from the table above, the makeup of the problem statement reasons is the same for all the problem statements. That makes sense since each reason enables the solving of the main problem. We show the *make lemonade* in italic to show that refresh yourself is the reason. Here let's show the model of the problem statement for Ms. Smith and Peter.

Given that each action identified above depends on communication, we can identify the actions themselves as function of communication. For example, we can think it like this. In order to refresh myself, I need to make a lemonade, *make a lemonade* is a function of communication. In order to make the lemonade, I need someone to provide me with lemon. In this case, *provide lemon* is also a function of communication. In order for me to refresh myself, I need someone to provide me with sugar to make the lemonade. Again, *provide sugar* is a function of communication. In this case, we can replace the actions by function of communication.



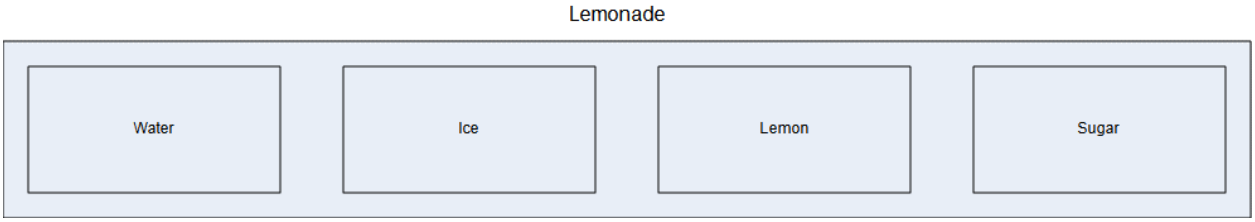
Here, let's show Peter's problem statement

Copyright 2023 *The Speak Logic Project*



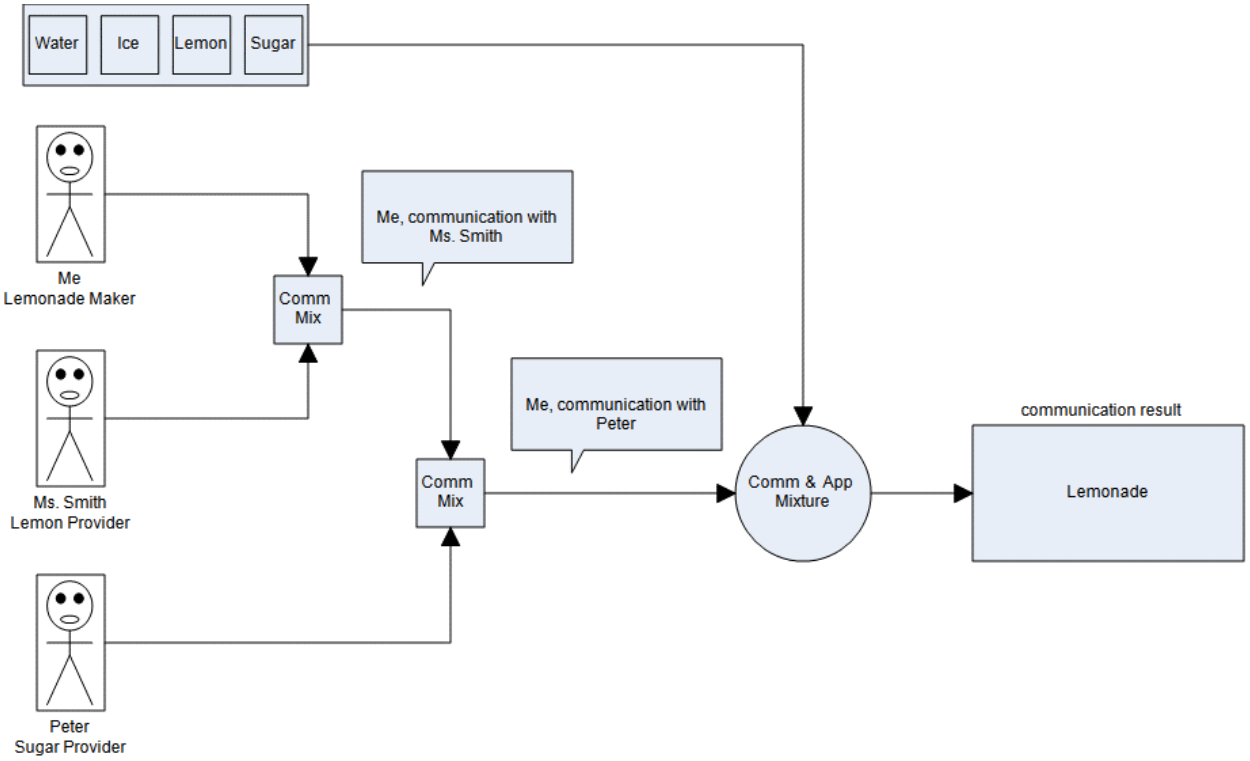
Partition of the Application

We know that the lemonade application is made of multiple parts of items. As we have shown, the lemonade application is made of water, ice, lemon, and sugar. Here, let's represent the lemonade application by the model below.



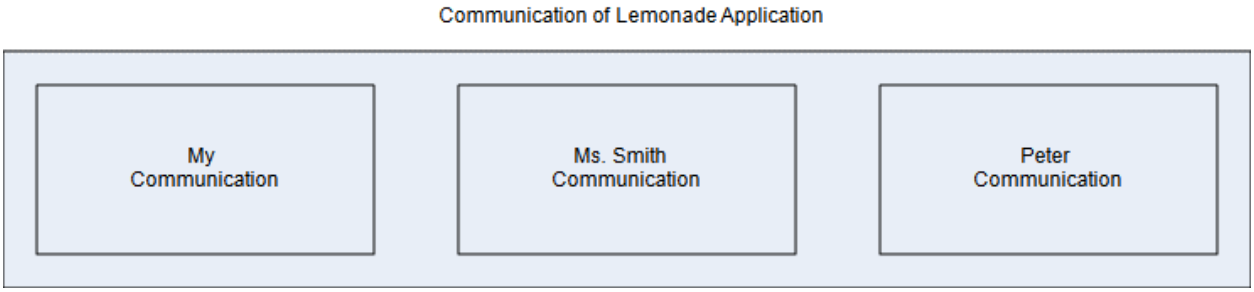
Now, let's show the model in relationship with people in the application and their communications.

Copyright 2023 *The Speak Logic Project*



Partition of the Application Communication

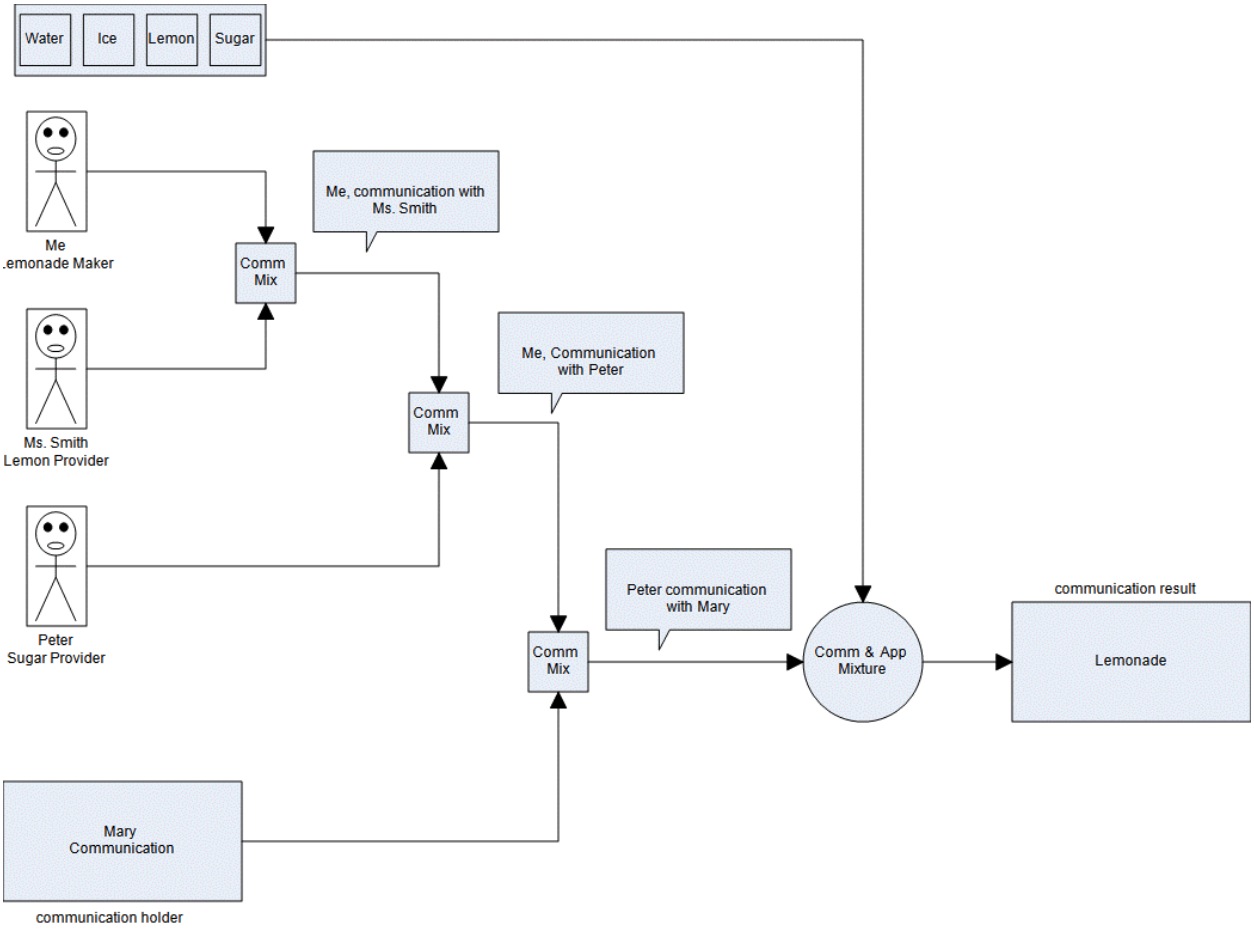
We know that the overall communication for the application include my communication, Ms. Smith's communication, and Peter's communication. In this case, we can represent the overall communication of the lemonade application by identifying each part of the communication as shown by this model.



In terms of people in the application and their communications, let's assume that Peter did not have the sugar, and he went to ask another person for that sugar. For example, when I contacted Peter, he told me to bring a cup, although he does not have the sugar, but he will get some from Mary. Here, we can use a communication holder to hold the communication for Mary. In this case, we can represent the model in the form below. The way to look at it, Mary is not

 Copyright 2023 *The Speak Logic Project*

physically presents in the application, however her communication enables the application. Therefore, we use a communication holder to hold her communication in the modeling of our application.



To better understand the process, let's represent the model above with the timelines of the communication between people in the application and respective communication holder.

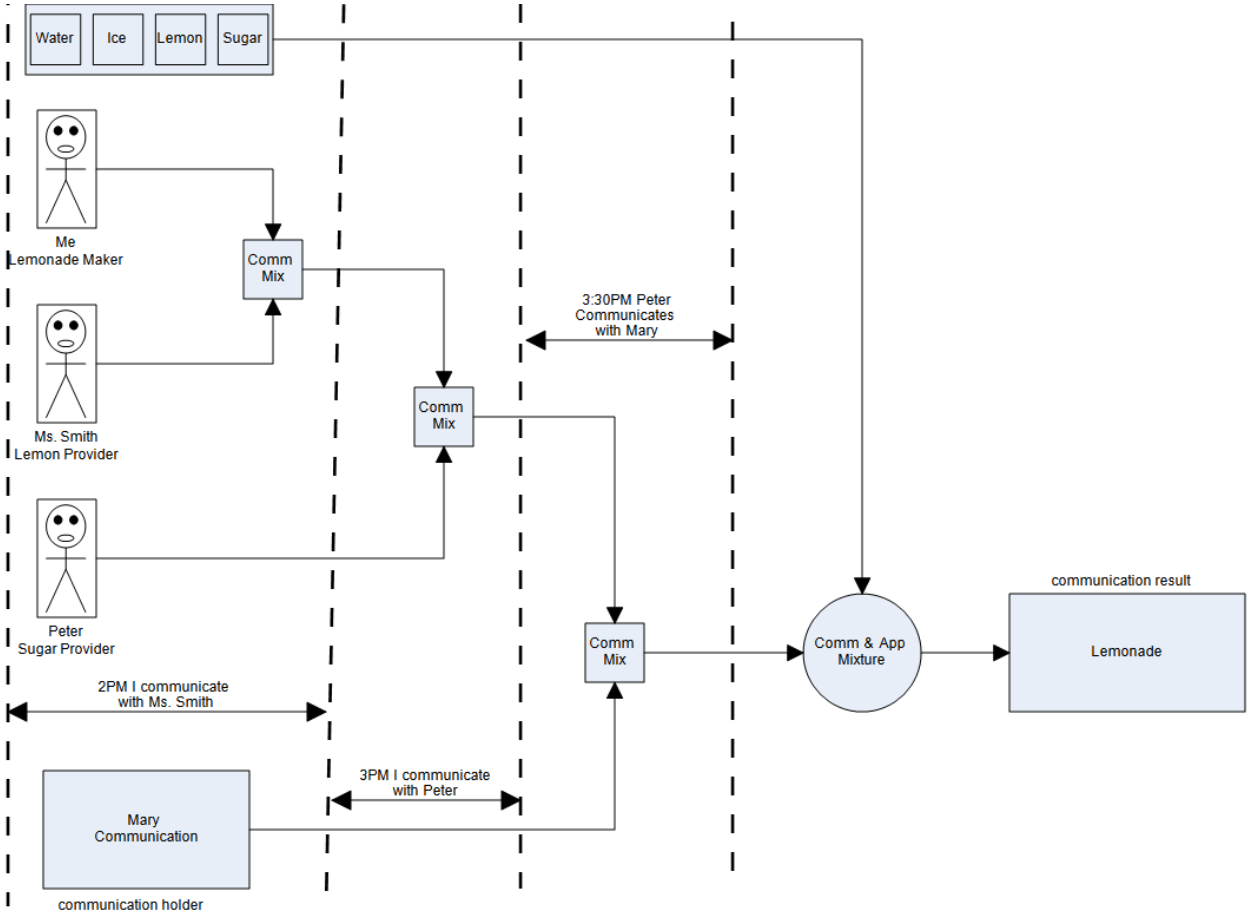
In term of function, by introducing Mary in the application, another function is also introduced in the application. In this case, the function of Peter is different than before. To better understand that, let's use the table below with some explanation.

People in Application	Function
Me	Make Lemonade
Ms. Smith	Provide Lemon
Peter	Receive Sugar

Copyright 2023 *The Speak Logic Project*

Mary (Comm. Holder)	Provide Sugar

The way to look at it, given that Peter does not have the sugar for the lemonade, he receives the sugar from Mary in order to make the lemonade. Mary is still the person who provides the sugar for the lemonade, therefore she still has a function in the application.



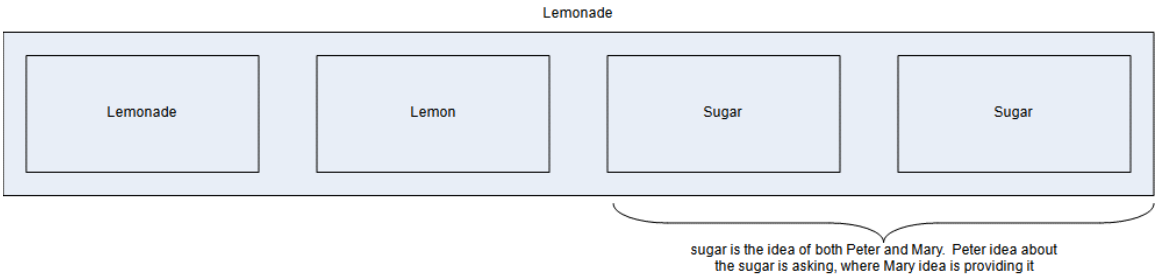
Below is another model where we identify both the communication function and the communication result. As separate entity, make lemonade is viewed as the function of the communication where the lemonade itself is the communication result.

External Communication in our Application

In term of idea of the application, by introducing Mary as a communication holder in our application, she also has an idea or her own idea about our application. In other words, Mary's idea is also a part of our application, as well as her communication. Let's show that in the table below with some explanations.

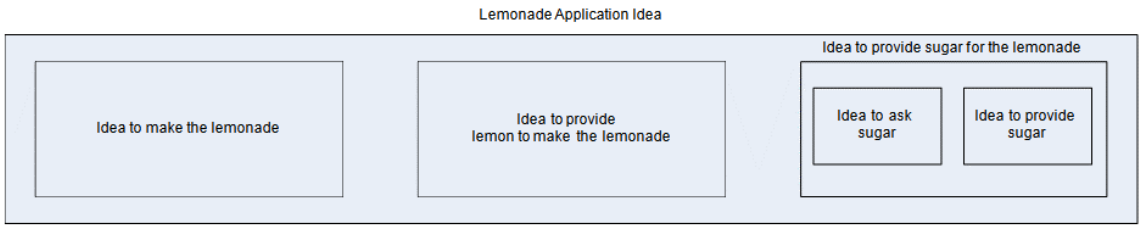
Copyright 2023 *The Speak Logic Project*

People in Application	Ideas
Me	Lemonade
Ms. Smith	Lemon
Peter	Sugar (idea of asking Mary sugar for the lemonade))
Mary (Comm. Holder)	Sugar (idea of providing sugar for the lemonade))

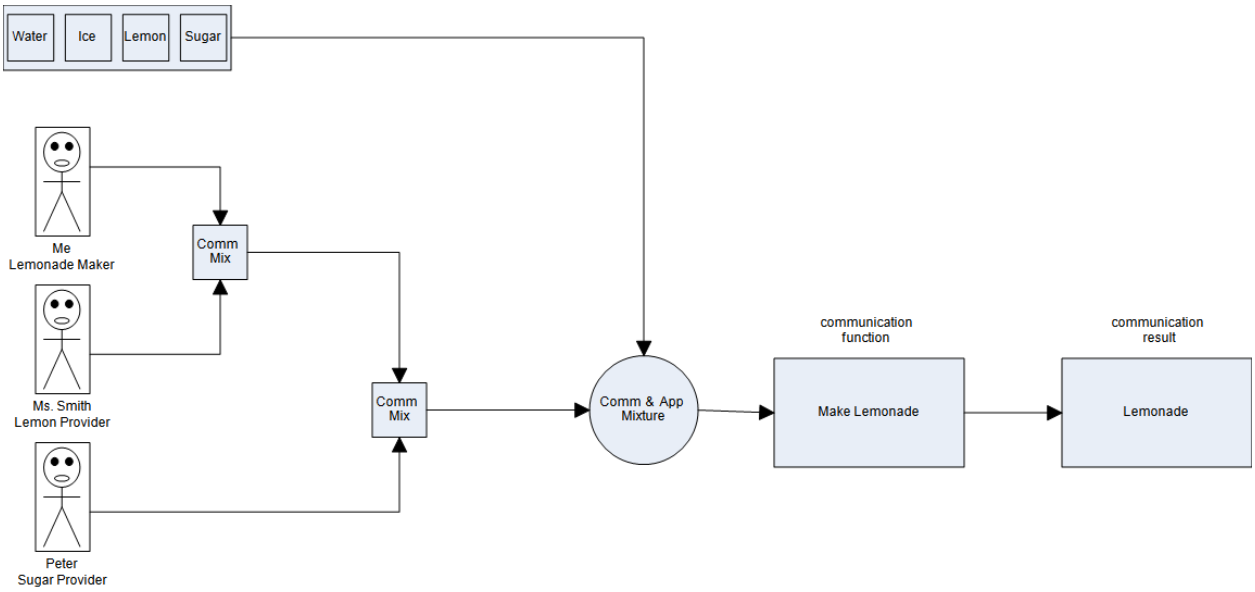


From the model above, the idea of the main application which is lemonade is from me, where the idea of lemon for the application is from Ms. Smith, the idea of sugar for the application is from Peter, where Mary also has an idea of the application which is providing the sugar for the application. While we don't use verbs to show them, we can put it like this: idea about making the lemonade, idea about providing lemon to make the lemonade, idea about getting sugar to make the lemonade and idea about providing sugar to make the lemonade.

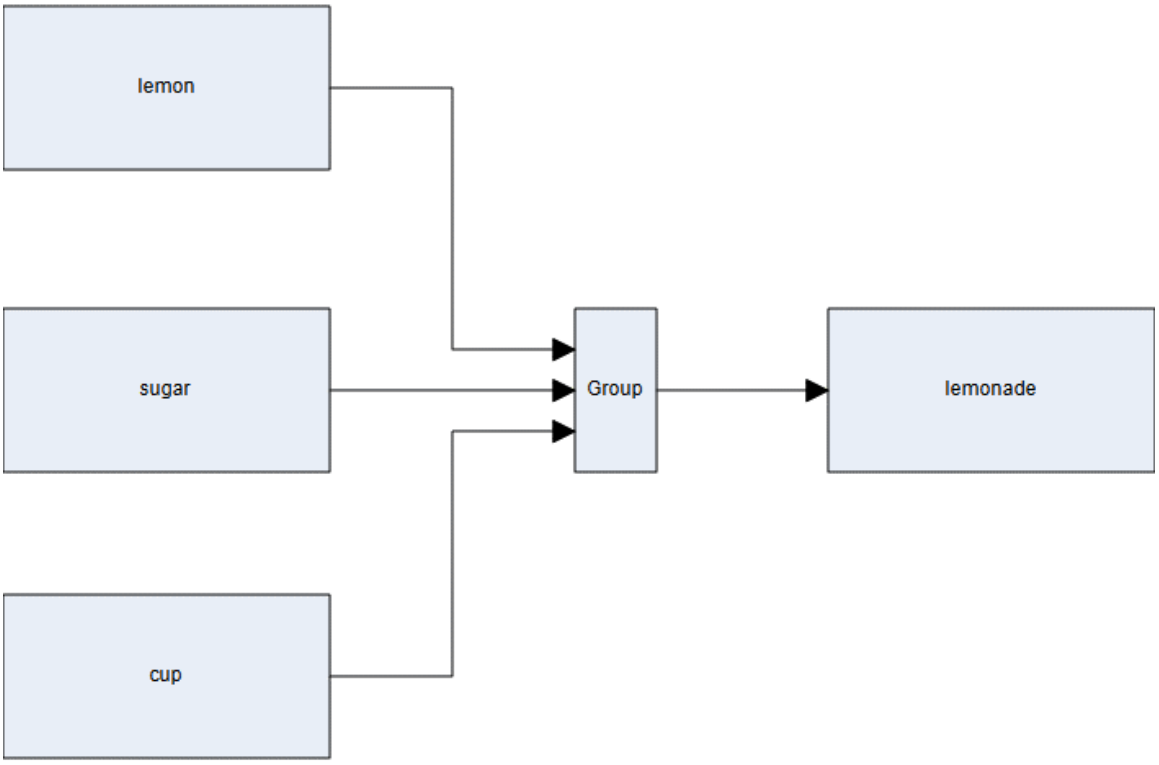
To better understand the model above, let's take it like this with additional explanations. We can think the lemonade application include idea to make the lemonade, idea to provide the lemon, and the idea to provide the sugar. Where the idea to provide the sugar include the idea to request or ask for the sugar and the idea to receive the sugar. Let's represent it again below with more explanation



 Copyright 2023 *The Speak Logic Project*

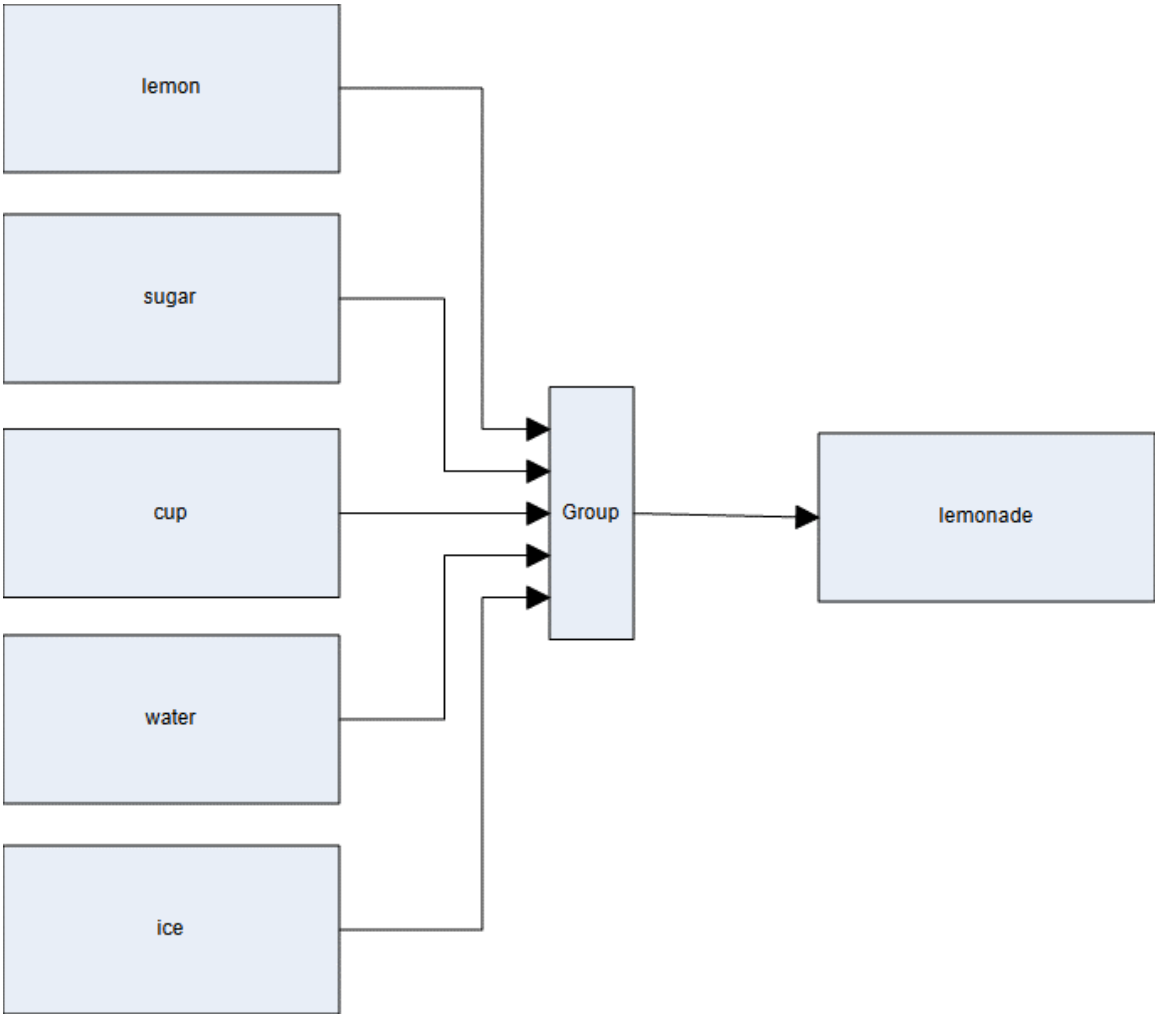


While it is not important, since our function is function of communication, in term of communication we can show a list of entities that we use to produce the result. For example, our communication enables to get lemon from Ms. Smith, sugar from Peter, and so forth.



Copyright 2023 *The Speak Logic Project*

Now if we want to, we can show all the entities that we use to produce the result of the application

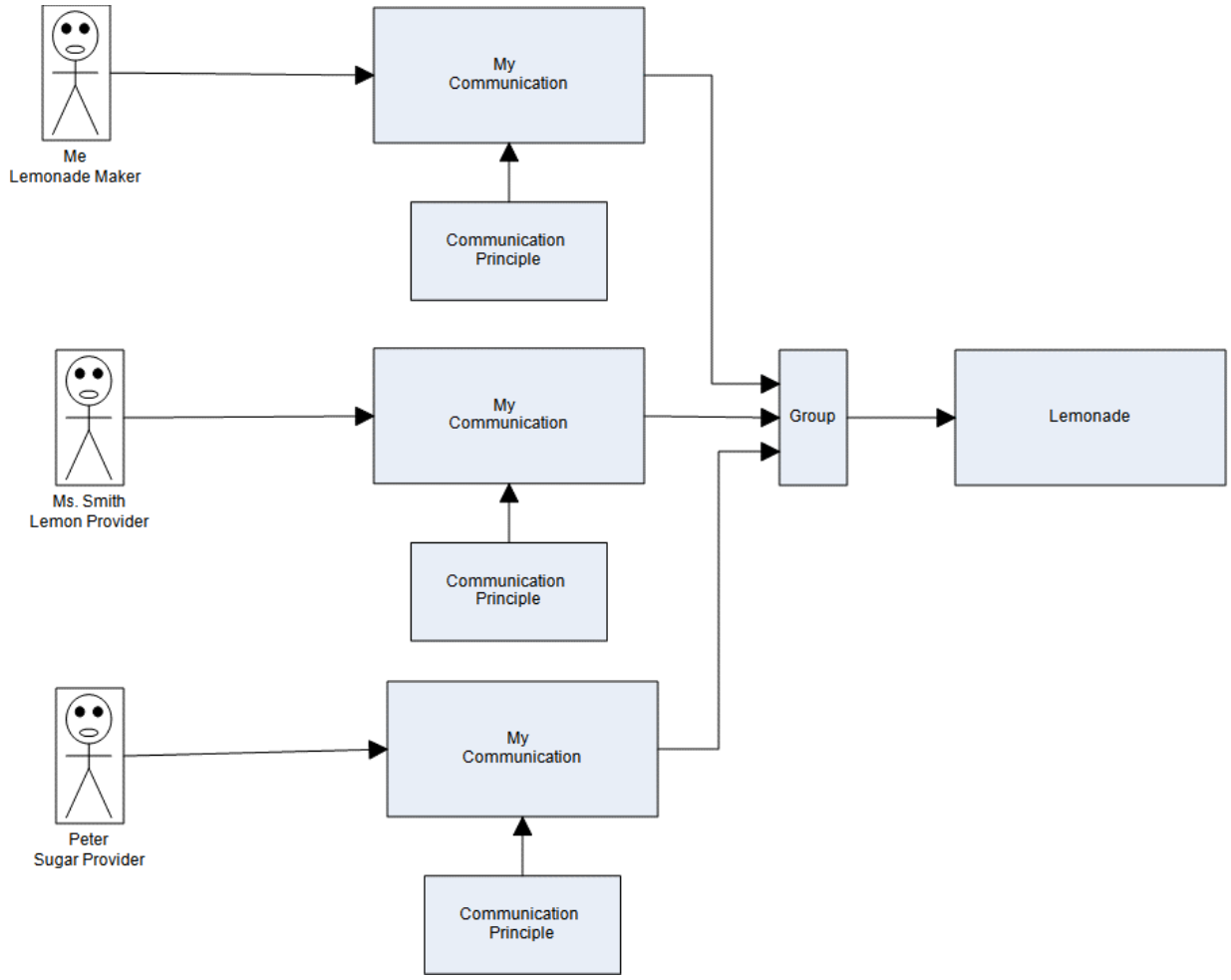


Given that we use principle of communication to communicate in our application, we can show that in the model as well. For example, I used the communication principle to communicate with Ms. Smith, and Peter, where both Ms. Smith and Peter also use the communication principle to communicate with me. In this case, we can show the model in the form below.

Applying the Principle in our Application

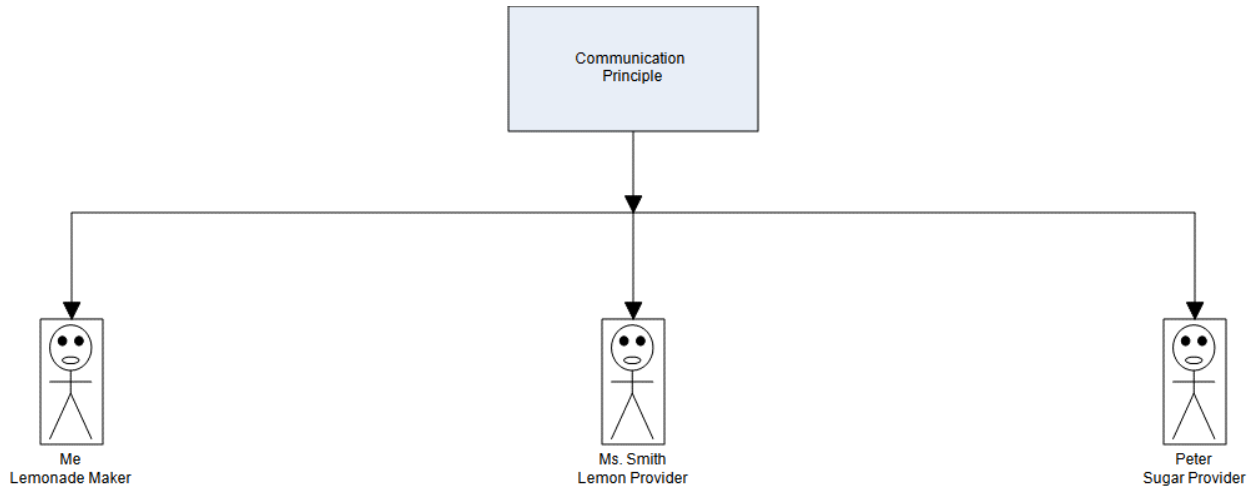
The way to look at it, the idea about making the lemonade is a part of making the lemonade. The idea about providing lemon to make the lemonade, is a part of making the lemonade. The idea about providing sugar to make the lemonade is a part of making the lemonade.

Copyright 2023 *The Speak Logic Project*



As shown above, each person in the application use the principle of communication to communicate about the application. In this case, we can say that the principle of communication enables the application. For instance, as a result of the application of the principle of communication, I was able to get lemon from Ms. Smith and sugar from Peter to enable me to make the lemonade. From what we have just said, we can put the principle on top of everybody as shown by this mode.

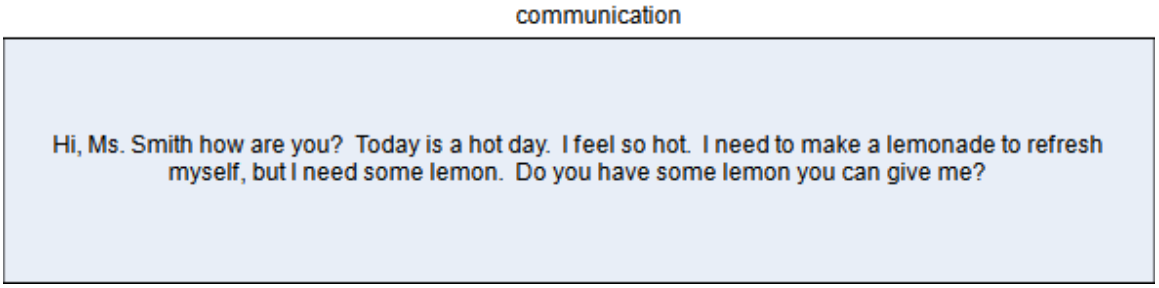
 Copyright 2023 *The Speak Logic Project*



The model above can be viewed as the principle of operation model, since it allows the making of the lemonade. For instance, by using the principle of communication, I was able to make the lemonade. In this case, the principle of communication is being viewed as the principle of operation of the lemonade. We can also say that the principle of operation of making the lemonade.

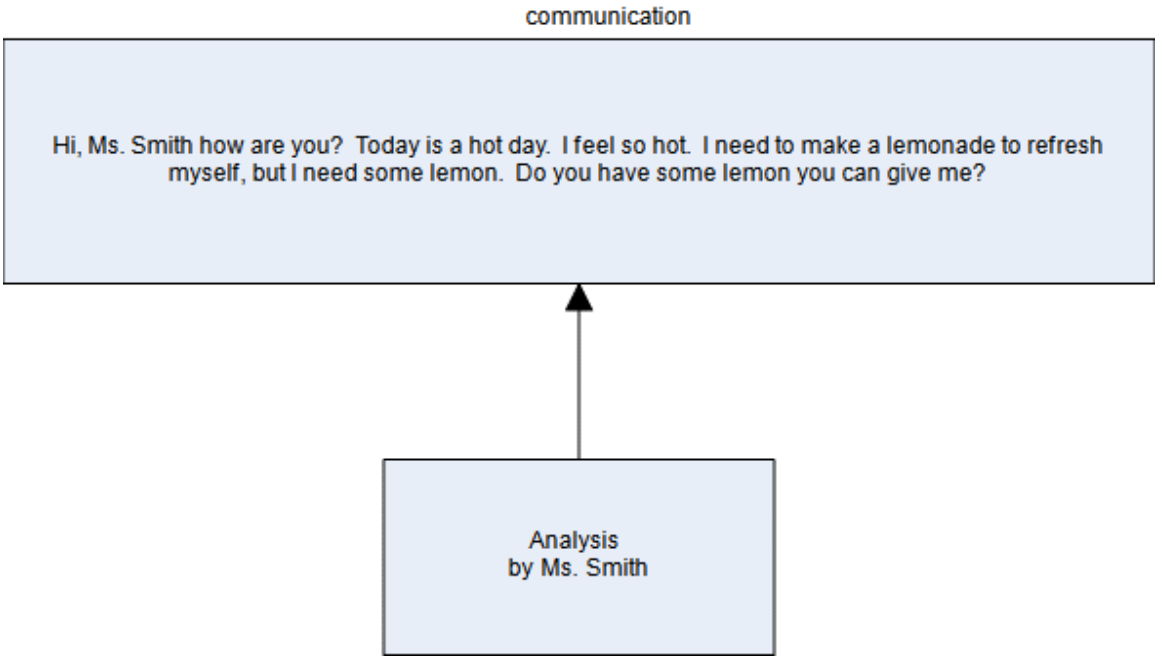
Identifying the Principle in our Communication

Now, let's try to identify the actual principle. The way to look at it, we use the principle of communication in our communications. From our communications, we can identify the principle of communication. Let's identify the communication with Ms. Smith for example.

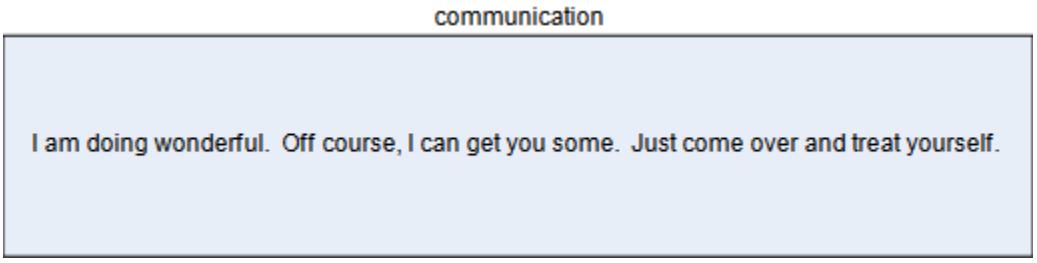


To understand the communication, Ms. Smith performs an analysis of the underlying communication. Let's represent both the communication and the analysis by the model below.

 Copyright 2023 *The Speak Logic Project*

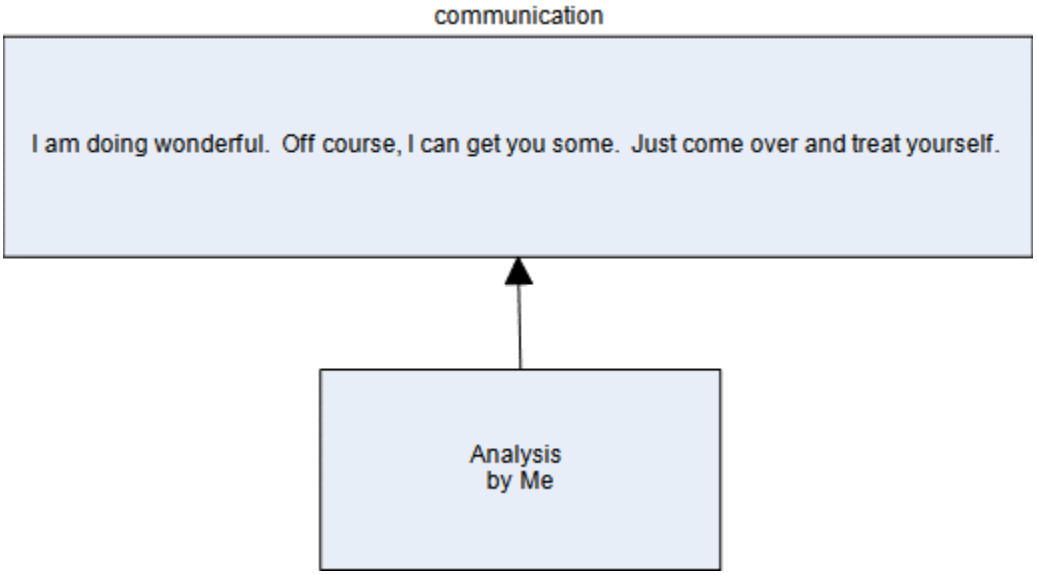


The model above shows that Ms. Smith analyzes the communication. After analyzing the communication, Ms. Smith replies to me with the following communication.



I perform an analysis of the communication as well to enable me to understand the communication. We can represent the process in this form.

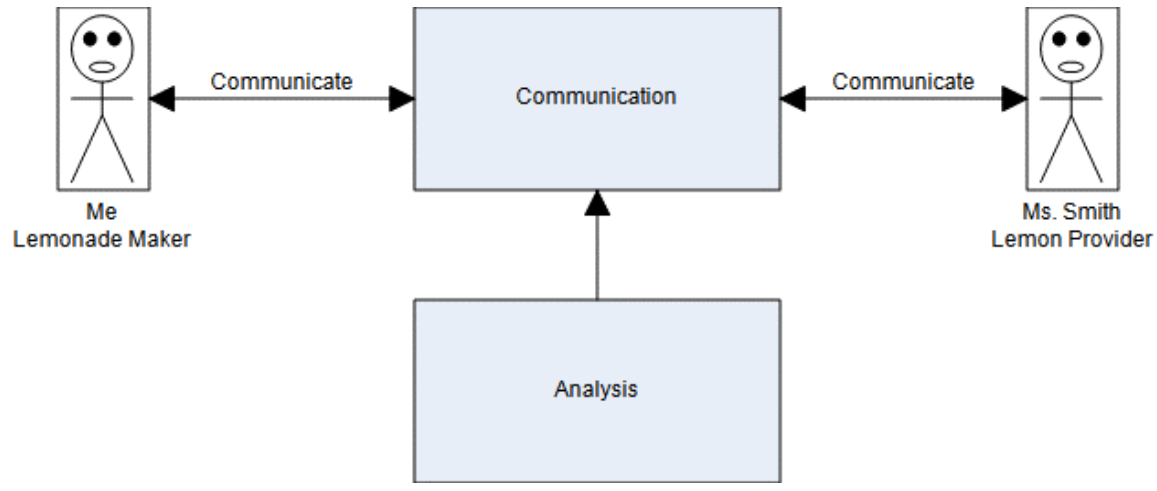
 Copyright 2023 *The Speak Logic Project*



What is important here, my communication to Ms. Smith is analyzed by Ms. Smith which enables Ms. Smith to understand it and Ms. Smith's communication is analyzed by me which enables me to understand it. Another way to look at it, given that we think before we communicate, my communication is analyzed by me before I communicate to Ms. Smith and Ms. Smith's communication is analyzed by her before she communicates with me.

Analysis of Communication

We can represent what we have just said above by this model. In this model, the analysis is being viewed as a separate entity from the communication, where I analyze my communication and Ms. Smith communication and vice versa.



 Copyright 2023 *The Speak Logic Project*

We know the principle of communication enables us to perform the analysis of the communication. In this case, from the communication itself, the principle can be identified. In other words, when we perform the analysis, we determine if the principle is included in the communication. Now, let's show the communications again and try to identify the principle in the communications.

My Communication	Ms. Smith Communication
Hi, Ms. Smith, how are you? Today is a hot day. I feel so hot. I need to make a lemonade to refresh myself, but I need some lemon. Do you have some lemon you can give me?	I am doing wonderful. Off course, I can get you some. Just come over and treat yourself.

From the communications above, we can identify the following principles. It does not matter the way we have them, this is simply an example. From the table, we identify the actual principles; we also number them.

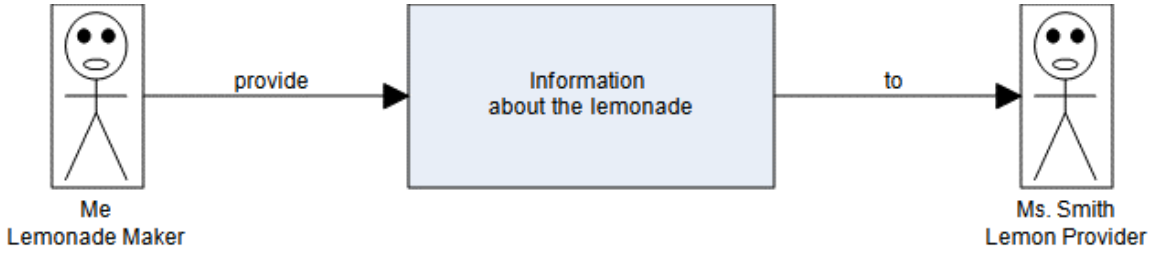
Principle Number	Principles
1	As a communication enabled entity, we must understand our communications in order for our communication to satisfy its objective.
2	During our communications, all parties must understand.
3	As a communication enabled entity, we use communication in what we do. During this process, all of us must understand the communication.
4	As a communication enabled entity, when we must understand each other communication that enables us to do what we do.
5	Within a given communication, there exists the communication and the principle itself.

From the example above, we have performed analysis under the underlying communications to help us identify the underlying principles. In terms of principles, we identify the principles above as the principles of communication.

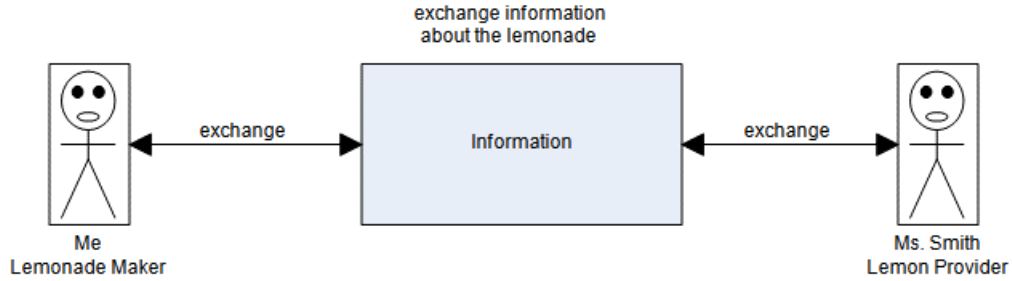
Now, let's analyze the communication again. I call Ms. Smith and tell her that I feel hot and I need to make a lemonade and I don't have lemon, she replies to me to come over and get some lemon. Now, if we analyze the process, we can see that during the communication with Ms.

 Copyright 2023 *The Speak Logic Project*

Smith, I provide information to Ms. Smith about the lemonade, and she also provides information to me about the lemon. Let's represent what we have just said by the model below.



We can also represent the model above in this form, since both of us exchange information about the lemonade and about the lemon.



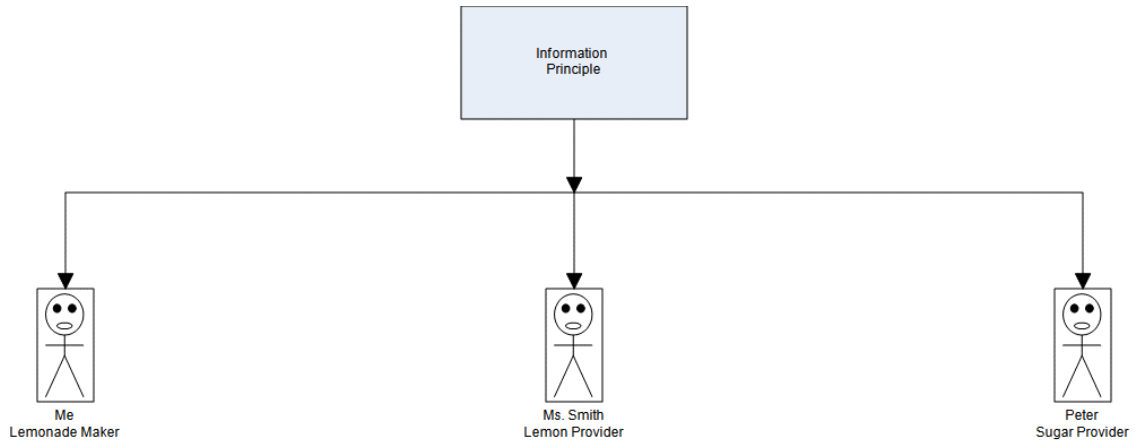
Previously, we have identified the communication principles, now we can analyze the information to identify the information principles. From what we have just said, we can identify some information principles from the underlying communications.

Principle Number	Principles
1	The information about an entity points to that entity
2	The information about an entity is determined by that entity
3	The correctness of information about an entity depends on that entity
4	Information is always presented in a form to enable us to understand it
5	During communication, words that we use to identify entities do not change the aspects of those entities. This is the same as saying that, given that we communicate by identifying entities by words, during communication our words do not change (cannot change) the aspects of those entities.

Copyright 2023 *The Speak Logic Project*

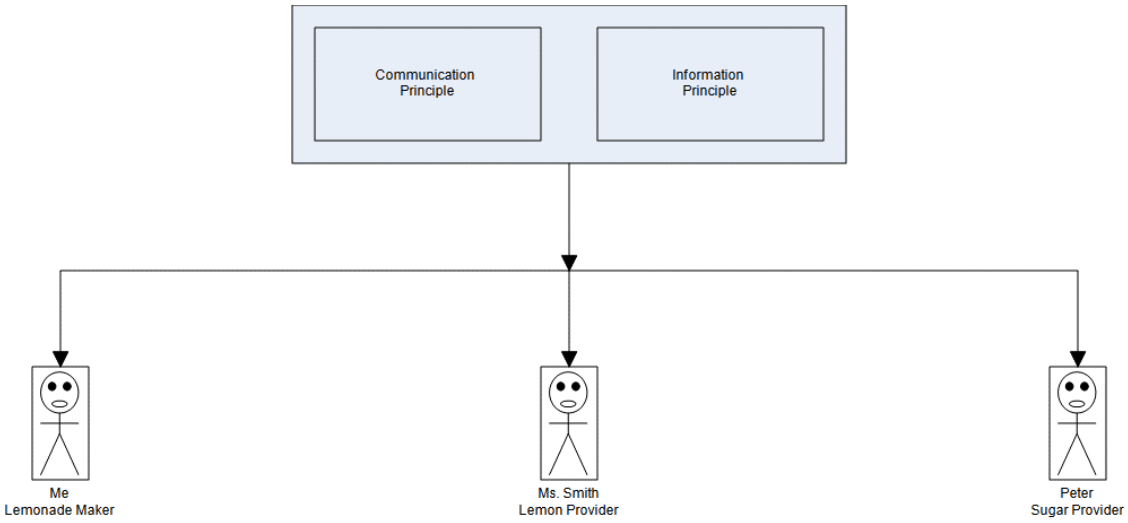
5	During communication, all entities preserve their original aspect (including definition and information).
6	Given that when we communicate we identify entities by words, during communication, those entities preserve their aspects.

As we can see form the table above, we have a list of information principles. I was able to use the information principle to get lemon and sugar from Ms. Smith and Peter to enable me to make the lemonade. In this case, we can see that we depend on the information principle as well to execute our application. From what we have just said, we can show the model, where we depend on the information principle to do what we do.

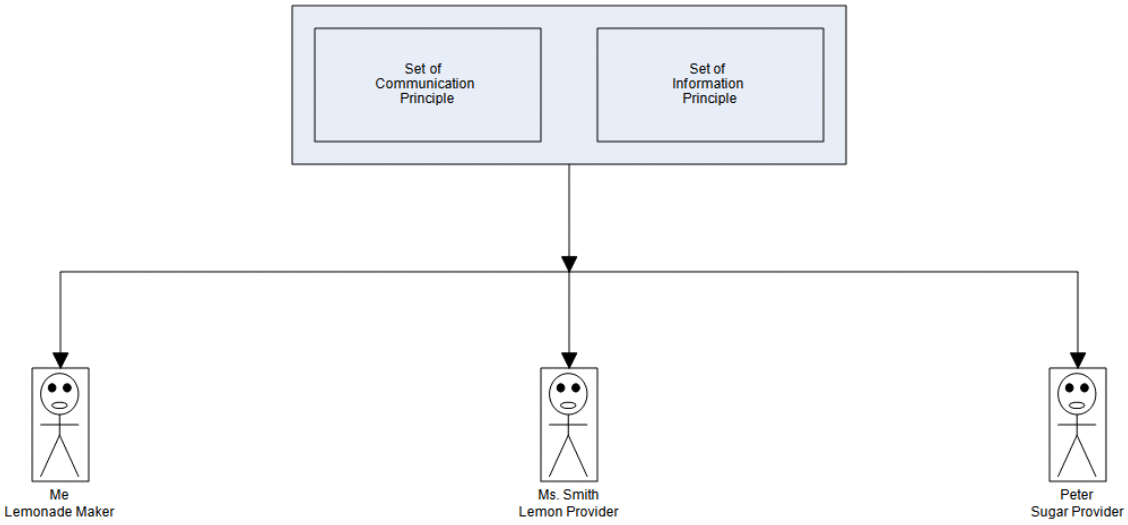


As we have stated and seen before, we depend on the communication principle to execute our application, we also depend on information principle to execute our application. We have represented both dependencies by the model above, as well as the one from the previous pages. Now, we can combine both of them together to show both principles in our dependency.

Copyright 2023 *The Speak Logic Project*



The model above shows that we depend on both the communication principle and the information principle to execute our application. While we denote the principles above as single entities, given that we have identified a list of both of them, we can represent them as set instead. For instance, we can represent the communication principle as set of communication principles and we can represent the information principle as set of information principles as shown below.



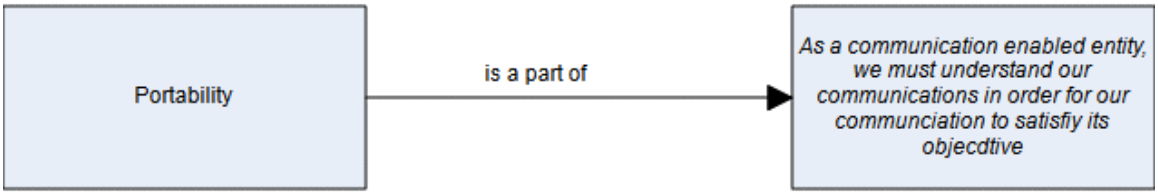
Principle Verification

We can use the aspect of the principle entity to verify that the principle entity is instead a principle. In other words, we have identified a principle, we can use the aspect of that principle to verify that it is a principle. We have identified a communication principle from the list, we can verify it by using the aspect of that principle.

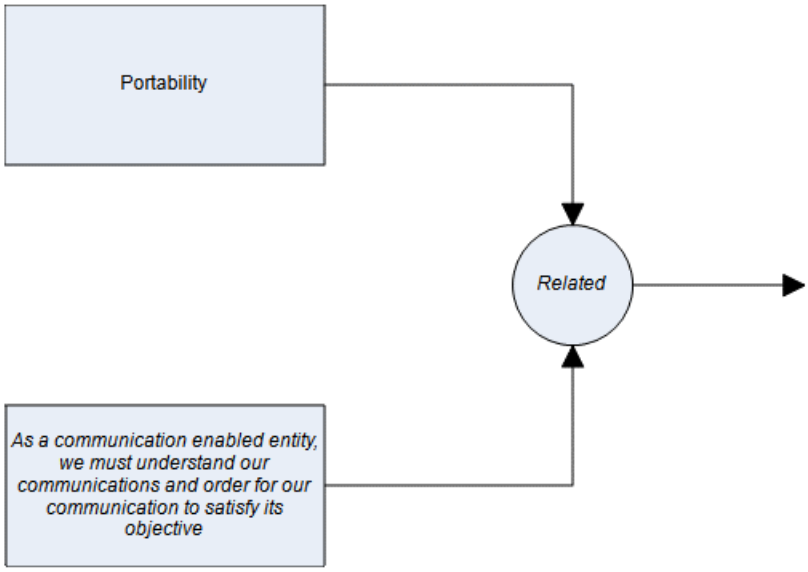
 Copyright 2023 *The Speak Logic Project*

We know that portability is an aspect of the principle entity. Now, let's use that aspect to verify one of the communication principles from the list. To do so, let's take the first principle from the list for instance. *As a communication enabled entity, we must understand our communications in order for our communication to satisfy its objective*.

We know that, the aspect of an entity is part of that entity and the aspect of an entity is related to that entity. In this case, portability is part of the principle entity and portability is related to the principle entity. Now, let's verify the principle by showing that portability is a part of it.

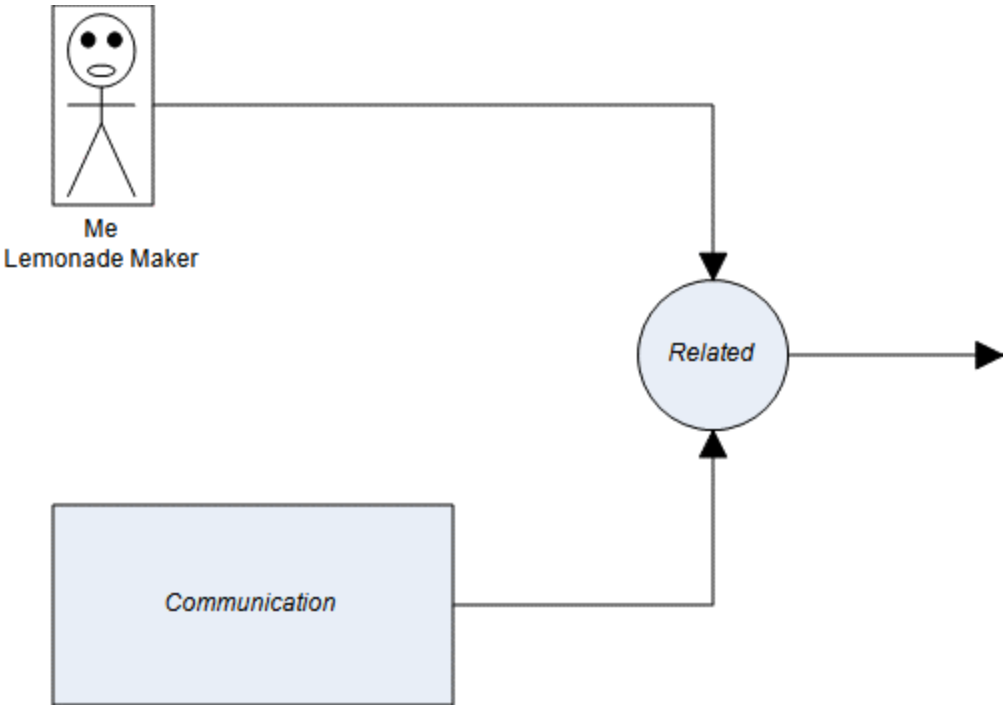


Since we said portability is related to the communication principle, we can put the relationship in the form below.

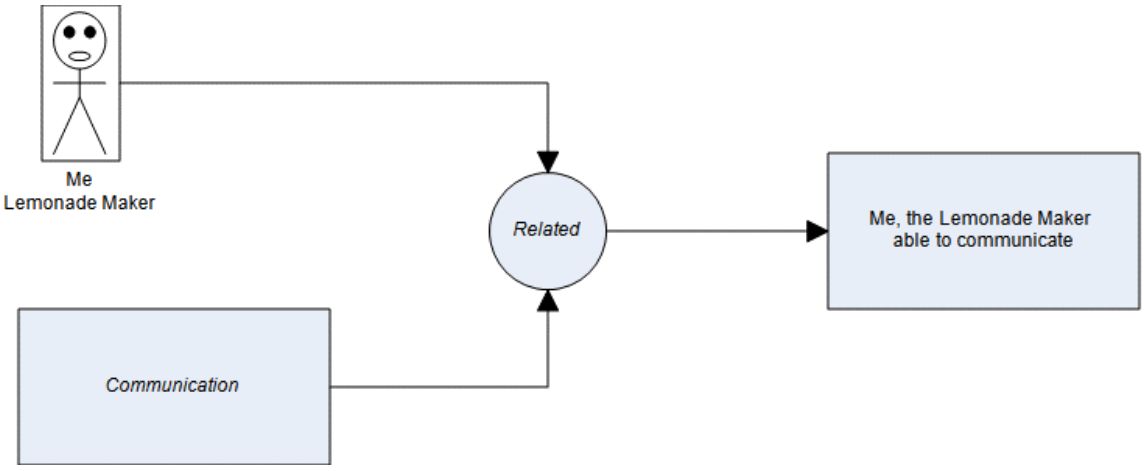


We know that communication is a part of us and it is considered as one of our aspect. In this case, we can represent that relationship in the form below.

 Copyright 2023 *The Speak Logic Project*

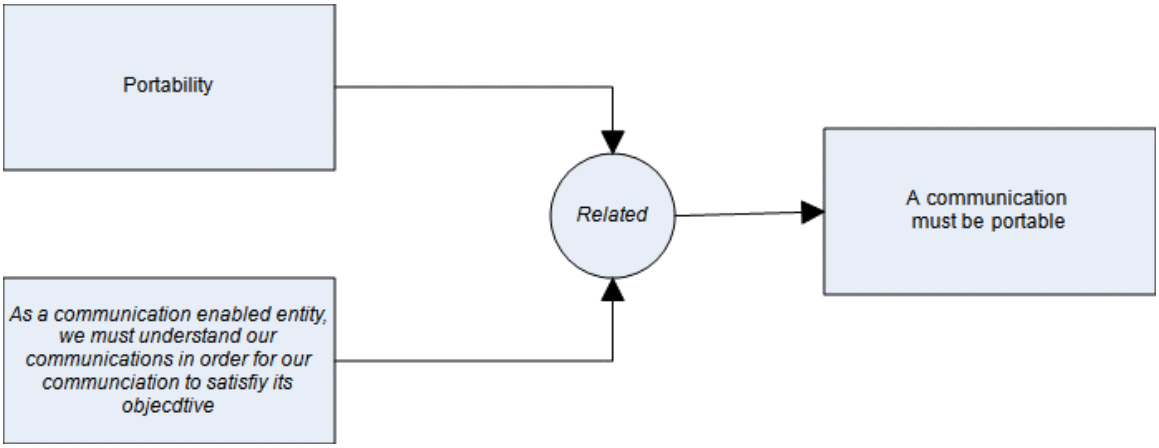


Now, we can combine the two relationship above which produce a result. We know that the relationship of two entities also produces an entity. Before we do so, let's show the resulted entity of each relationship above. For instance, let's look at the first one. Given that me, the lemonade maker is related to communication. In this case, the lemonade maker must be able to communicate. As we can see, the communication of the lemonade make is the result of that relationship. We can show it here and we can also verify it. The fact that the lemonade maker is able to communicate with Ms. Smith and Peter, thus that relationship below is verified.



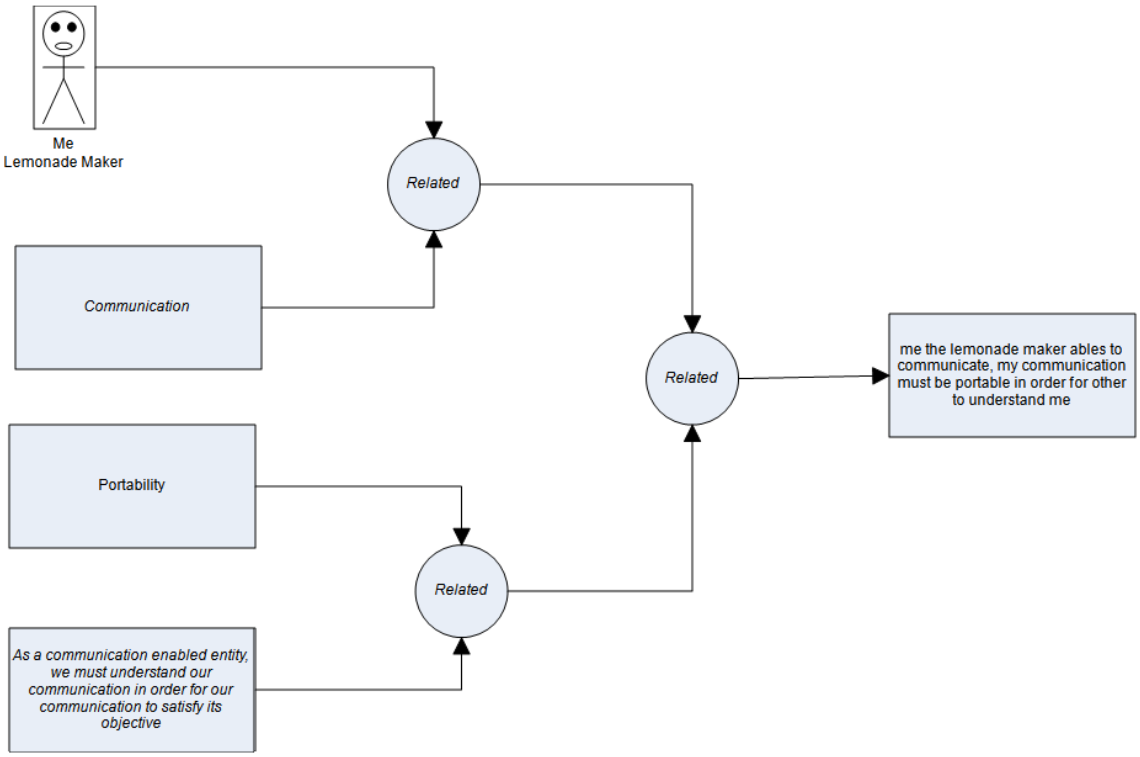
Below, we show a second relationship, the fact that portability is a part of communication, a communication must be portable

 Copyright 2023 *The Speak Logic Project*



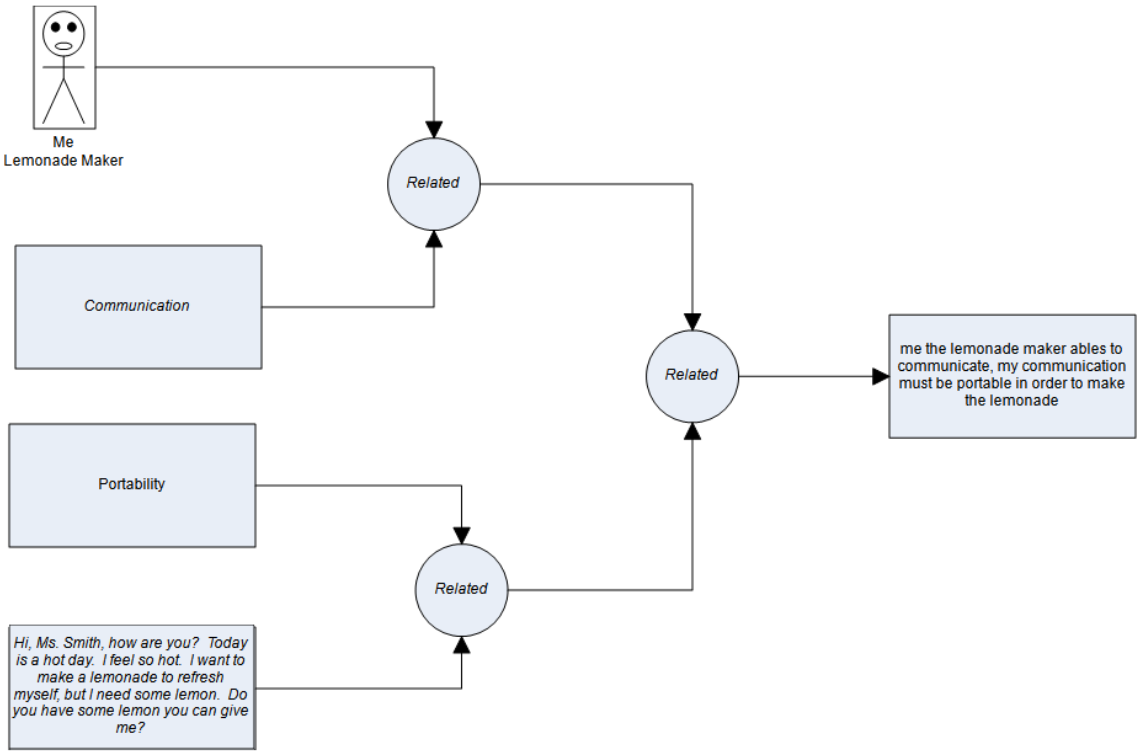
When we combine the two relationships above, we have the relationship below, which is the result of the relationship. The way to look at it, me the lemonade maker is able to communication with Ms. Smith and Peter about the lemonade. My communication must be portable, I must be able to communicate with Ms. Smith and Peter about making the lemonade. Able to communicate with Ms. Smith and Peter about the lemonade is the result of that relationship. In this case, I can call Ms. Smith and Peter and we can talk about making the lemonade and I can also go to their houses to talk to them about the lemonade and to get lemon and sugar to make the lemonade.

Copyright 2023 *The Speak Logic Project*



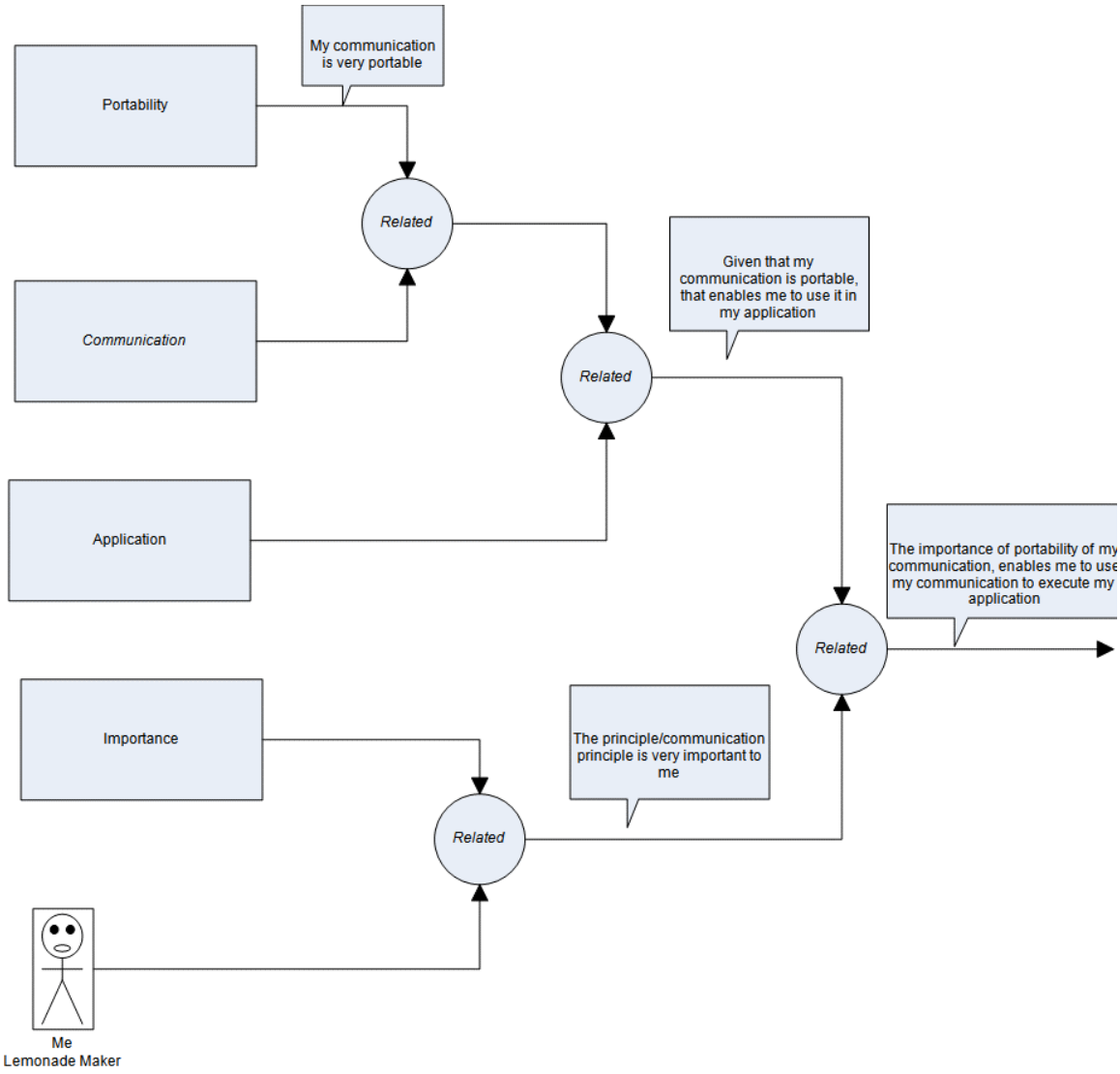
If we want, we can replace the principle by the actual communication, since the principle is already included in the communication. By doing so, we have the relationship below. The result is that our communication must be portable in order for us to execute our functions or to do what we do.

 Copyright 2023 *The Speak Logic Project*



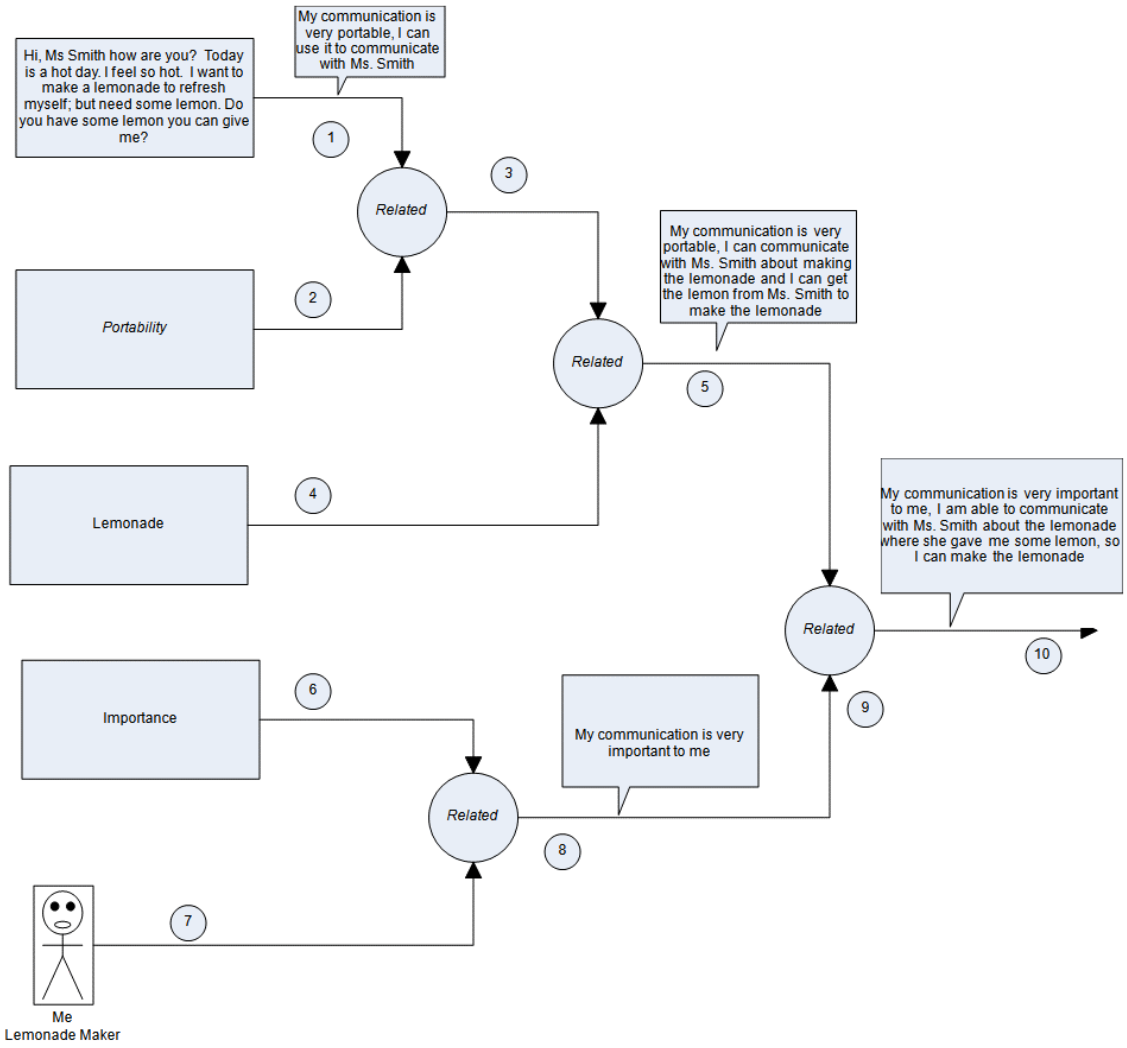
From the example above, we show that the relationship of the aspect of the principle, our aspect, as well as the result of that relationship. We know that the result of an identified relationship produces an entity and we have given multiple examples. Here, let's show another relationship.

 Copyright 2023 *The Speak Logic Project*



Now, we can use the actual communication to show that relationship in the lemonade application. Take the relationship at node 10 for example, given that all the aspects of the principle are related, if a communication is being viewed as important, that communication must be portable as well. The way to look at it, from the result, the portability of the communication is already been taken into consideration, although it is not mentioned in that relationship.

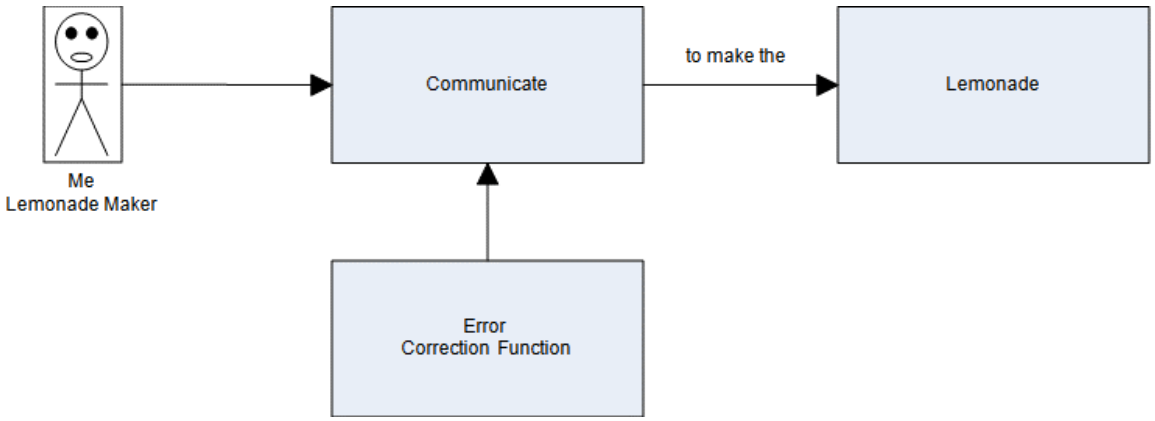
 Copyright 2023 *The Speak Logic Project*



Identifying Error in our Application

Now, let's take a look of errors in communication. As we can see, with the absence of errors in communication, I was able to make the lemonade to solve the underlying problem. If an error is introduced in the communication, it must be corrected in order for the application to execute without the error. Now, let's take a look of the process

Copyright 2023 *The Speak Logic Project*



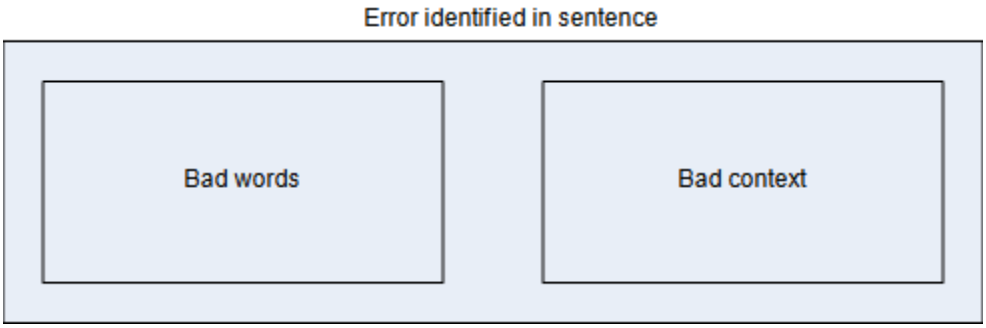
From the model above, if there is an error in my communication, it must be corrected in order for the application to execute properly. In term of error in communication, let's take a look of my communication with Ms. Smith and Peter.

My Communication to Ms. Smith	My Communication to Peter
Hi, Ms. Smith, how are you? I feel so hot today. I need to make a lemonade to refresh myself, but I don't have any lemon. Do you have some lemon you can give me so I can make the lemonade?	*Hi Peter, how are you? I feel so hot today. I need to make a lemonade, but I don't have any sugar. Do you have some you can give me?*

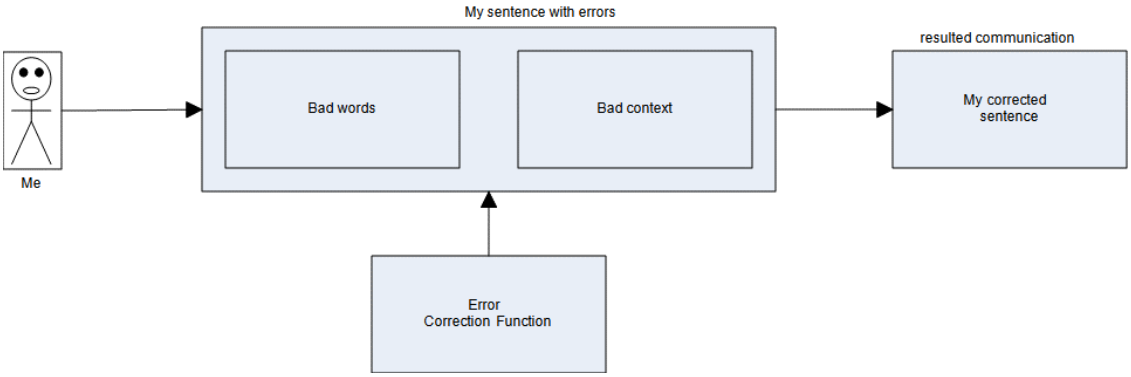
From the communications above, if there is an error, that will affect the execution of the application. For example, errors from the communications will prevent me from getting the lemon and the sugar to make the lemonade. By not making the lemonade, the underlying problem is not solved. To do so, the errors must be corrected.

From the underlying communications above, there are two ways to look at error in communication: first, the communications or sentences can include some bad words. Second, the communication/sentences may not make sense or may be in bad context. From what we have just said, the error in a sentence can be identified by the words as well as the context. Let's represent that below

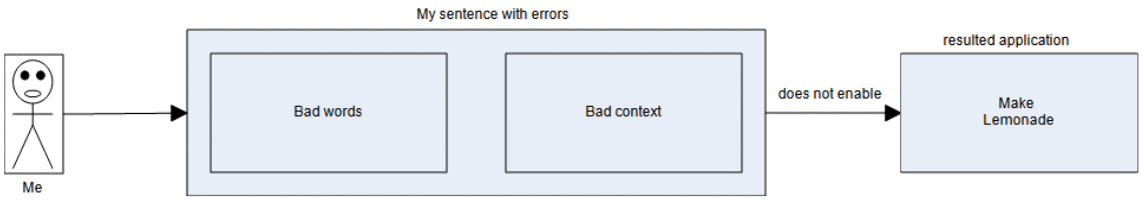
 Copyright 2023 *The Speak Logic Project*



From the model above, if we identify a sentence with an error, the sentence may have some bad/wrong words as well as bad or wrong context or does not make sense. The correction process enables the replacement of the bad words by good words as well as changing the context of the sentence. In other words, we replace words, we also replace context or sense of the sentence to make it makes sense. This is basically what the error correction function does, if there is an error in the sentence, the error correction function replaces each bad words as well as the context of the sentence. Let's represent the process by the model below.

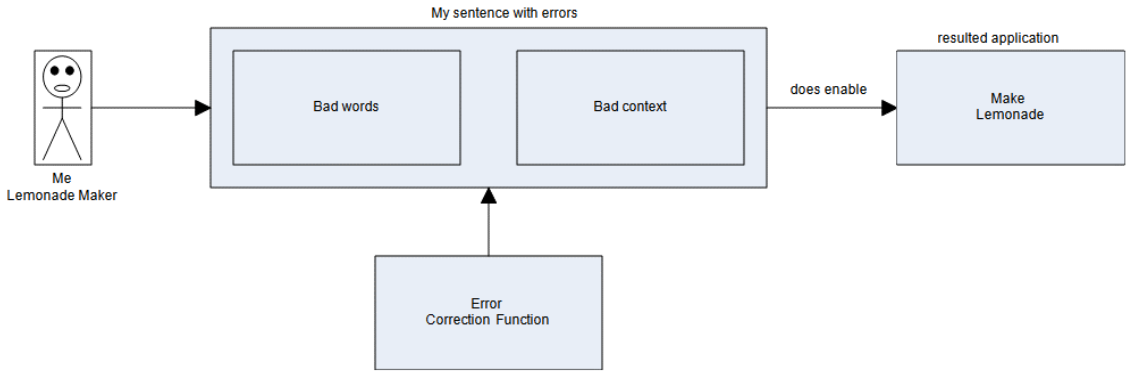


From the model above, the error correction function corrects the underlying sentence by replacing bad words with good words as well as changing the context of the sentence. The result enables the execution of the application without error. Here, let's represent it in two ways in terms of the application. My communication with errors, does not enable me to make the lemonade while my communication without errors enables me to make the lemonade.



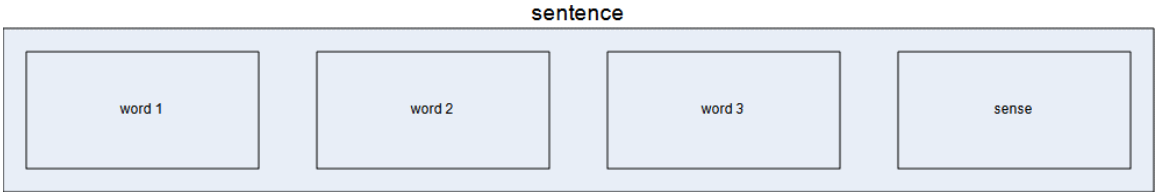
Copyright 2023 *The Speak Logic Project*

With the corrected communication by the error correction function, my communication enables me to make the lemonade.

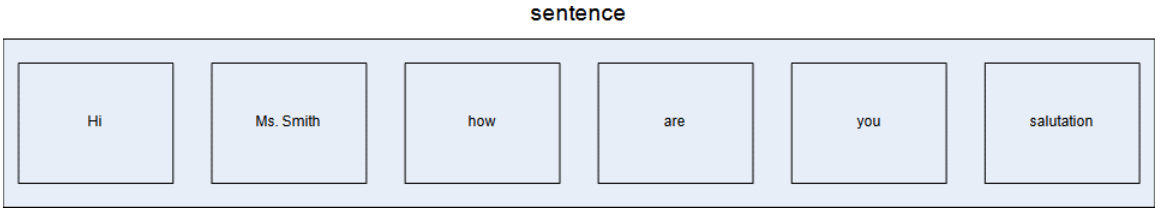


The way to look at it, even without a model to show the error correction function, an error free communication is viewed as a corrected communication. In this case, we can say that in an error free communication, the error correction function is naturally applied or has already been applied naturally to that communication. Therefore, there is no need to show that in a model where the communication contains no error.

Another way to look at it, a sentence is a collection of words or a group of words. We can think a sentence as a group of words plus context or sense of that sentence. In this case, we can represent a sentence in this form

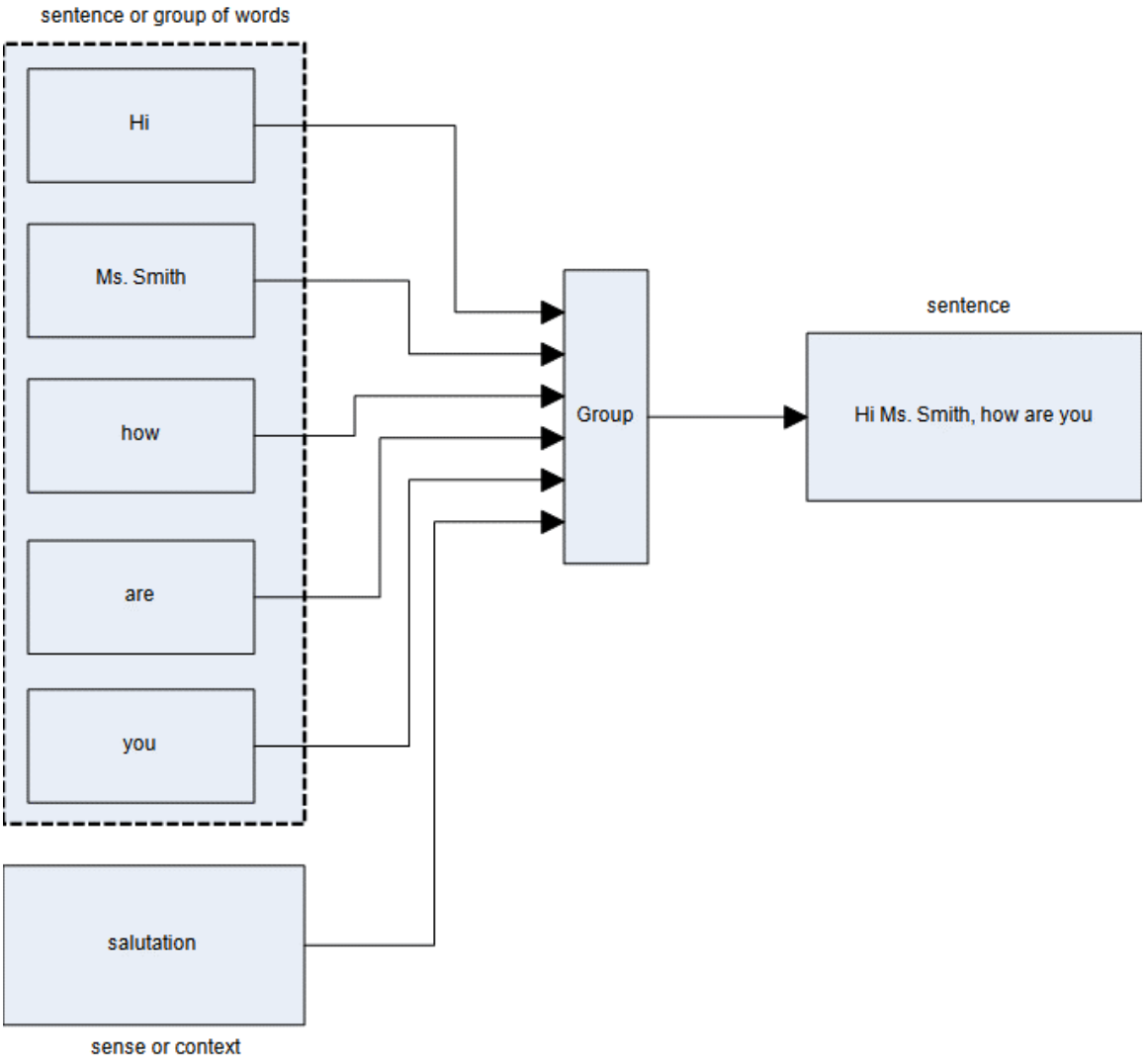


Let's represent my communication to Ms. Smith as an example, *Hi Ms. Smith, how are you?*. This sentence can be represented by grouping each word in the sentence plus the sense of the sentence or the context of the sentence. Let's represent that in this form below.



 Copyright 2023 *The Speak Logic Project*

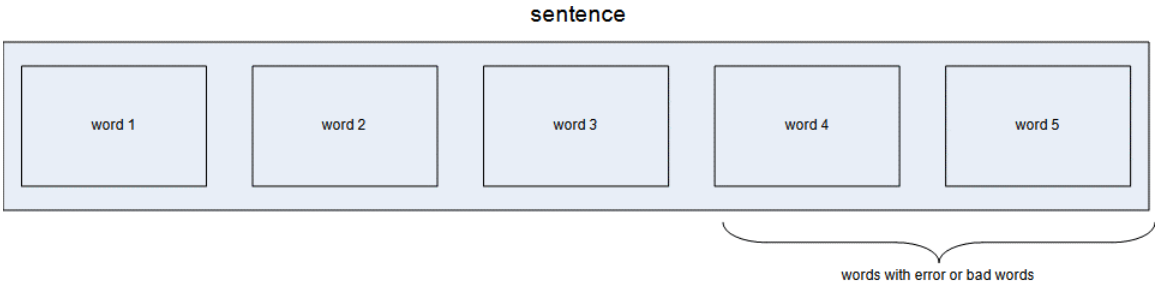
From the model above, we group all the words in the sentence and we replace the sense of the sentence by its own entity. To better understand that, let's represent it in this form



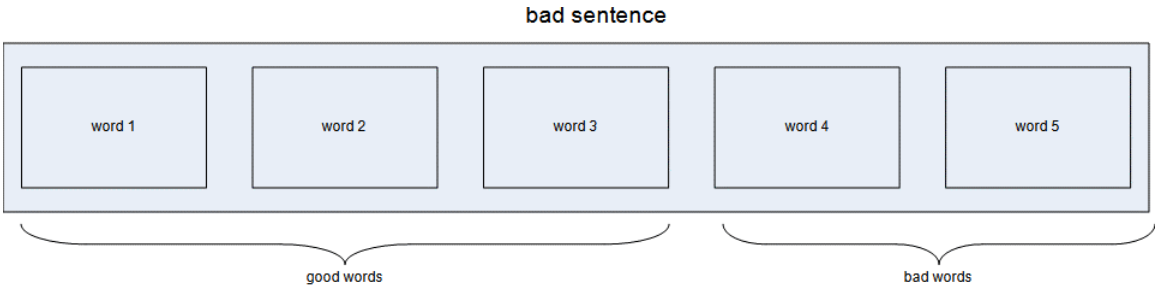
Correction of Error in our Application

From the model above, the sentence is made of five words, plus the context of the sentence. By taking error in communication into consideration, the errors can happen in the words in the sentence as well as the context of the sentence. Let's take error in communication where we have 5 words in the sentence, errors can be identified in some words in the sentence. In term of error in communication, let's represent the sentence with error, where there are errors in some words. In this case we have

Copyright 2023 *The Speak Logic Project*

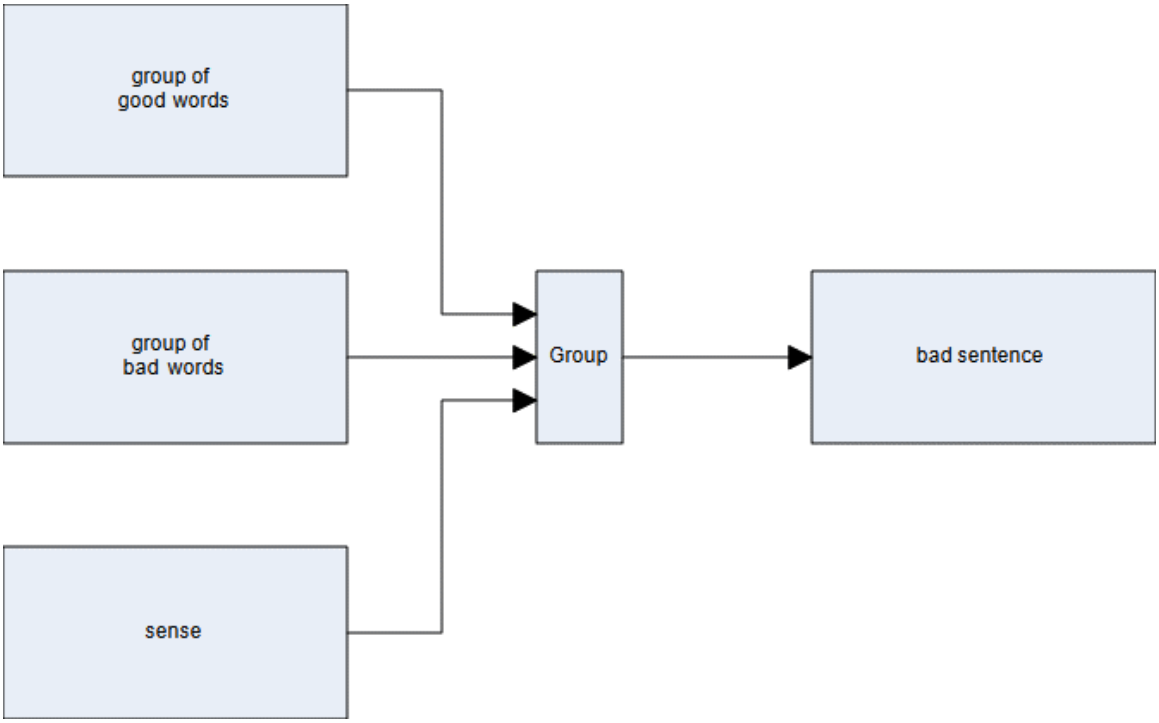


Given that error in the sentence produces a sentence with error, we can call that sentence a bad sentence. From the model above, our sentence has two groups of words, one of them is good and the other one is bad. We can identify the good one as group of good words and we can identify the bad one as group of bad words. From what we have just said, we can represent that in this form

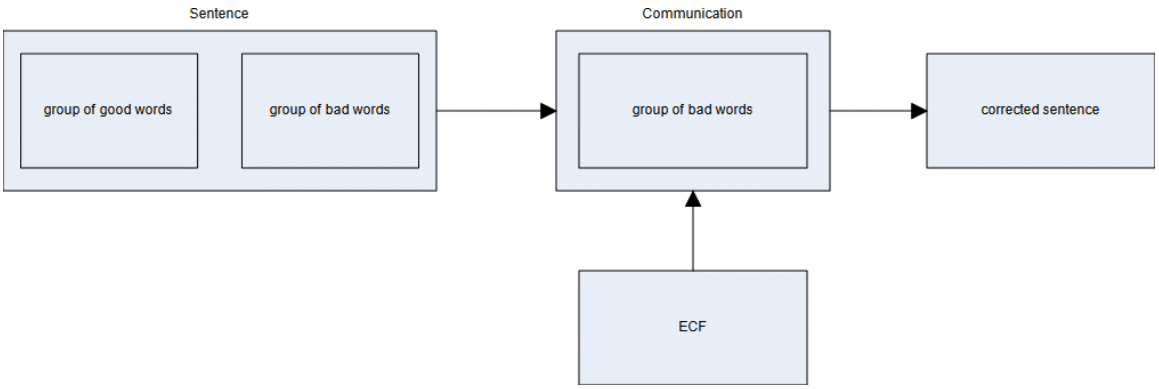


When a sentence is bad, the context of that sentence is also bad. Here, let's represent the groups in this form with the inclusion of the context of the sentence.

 Copyright 2023 *The Speak Logic Project*

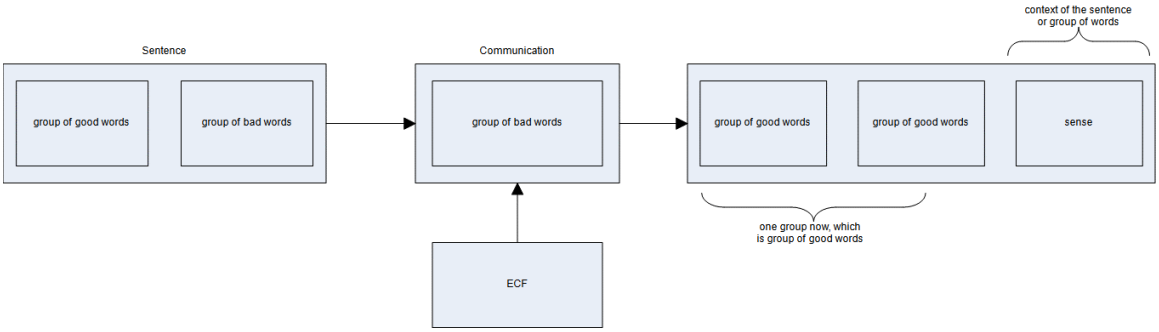


In term of correction, by using the error correction function, the group of bad words must be corrected in order to remove error from the sentence. From what we have just said, we can represent the error correction function that performs the correction with the group of bad words.



As we can see, the error correction function disregards the group of good words and perform the correction in the group of bard words which produces a corrected sentence. For instance from the bad sentence above, the error correction function performs the correction of word 4 and word 5 by replacing them with good words in order to correct the sentence. The error correction function also adds sense to the correction to complete the correction process. To better understand the process, let's show this model

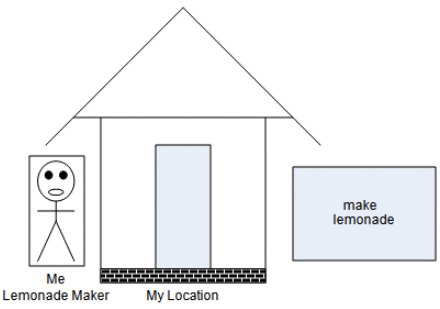
 Copyright 2023 *The Speak Logic Project*

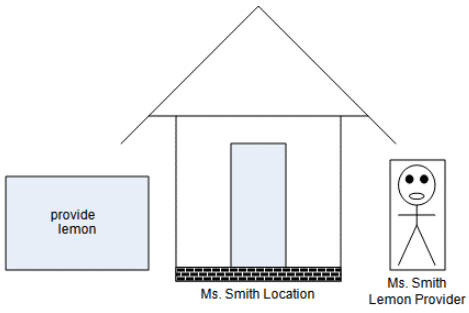


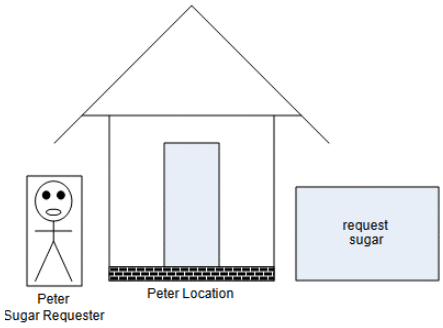
From above, the error correction function replaces the bad words in the sentence then add sense to the sentence to finalize the correction process. The resulted sentence is always error free.

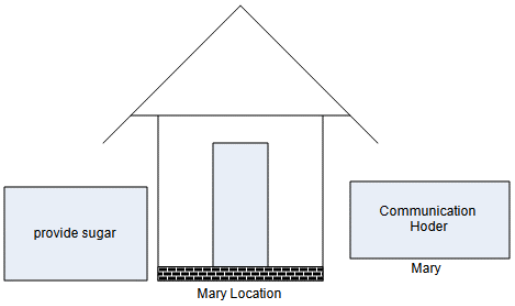
Application Execution Location

If we want to, we can provide the locations where the functions are executed. For example, I make the lemonade at my location which is the location where I execute my function. Ms. Smith provides the lemon at her location, Peter request for sugar at his location, and Mary provides sugar at here location. Here, we can show that in the form below where each person executes specific function of the application at his/her respective location.







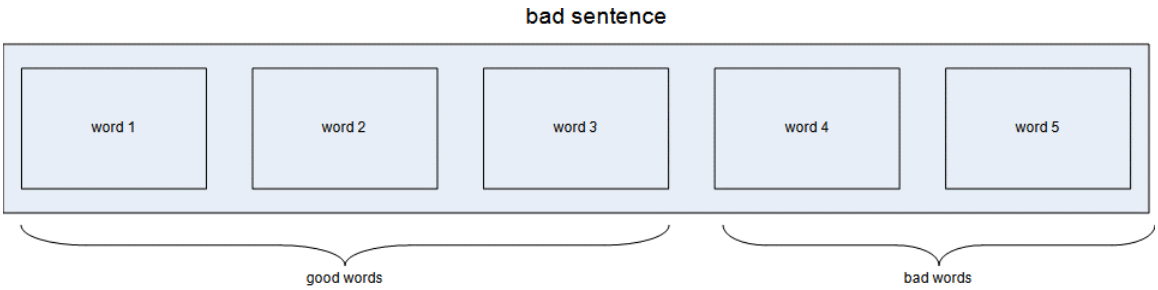


Copyright 2023 *The Speak Logic Project*

Understanding the Correction Process

To better understand the correction process, let's take like this. This can also be viewed as a summary of what we have learned previously. To better understand that, we identified the overall communication of the application as single sentence. Where that sentence is a group of words. With the introduction of error in communication, that sentence is viewed as sentence with error.

With error in that sentence, the group of words can be viewed as two groups: one with error and another one without error as shown below.



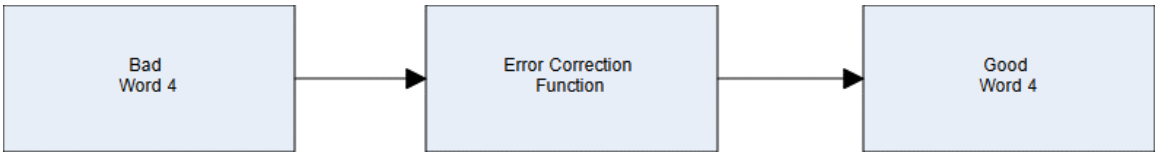
In order to cored error in the sentence above, we apply the following principles. We already know that a communication that contains error is viewed as a communication that does not include the principle. In other words, a sentence that contains error is viewed as sentence which lacks of the principle of communication. Here, let's identify some of those principles.

Principle Number	Principles
1	If all words in a sentence are correct, they must point to the right entity
2	In a given sentence, each words must point to its identified entity
3	In a given sentence that makes up of word 1, word 2, word 3, word 4 and word 5, if the sentence is correct, they must exist a similarity between the words. This is the same as saying that, if a sentence is correct, there must be a relationship between words in that sentence
4	If a given sentence is correct, the entities the words point to must exist. In other words, if the words are correct, the entities the words identify must exist and the words must point to them.

Copyright 2023 *The Speak Logic Project*

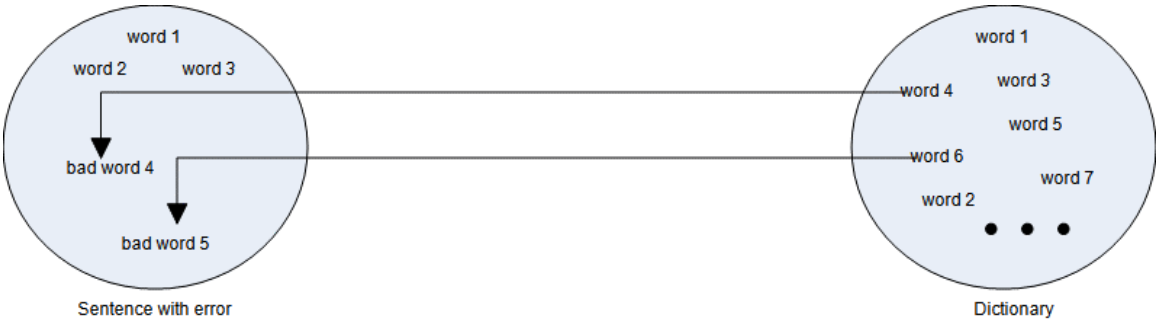
Base on the principles above, we use the error correction function to perform the analysis on the group of bad words in the sentence. In this case, we perform our analysis in a word by word basis.

For example, let's consider my communication with Ms. Smith as a single sentence that makes up of 5 words, were the last 2 words are bad where the first 3 are good. In this case, we can perform the analysis in the last two words which are word 4 and word 5. Now, we can apply the error correction function on word 4 as shown below.



By applying the error correction function on the first bard word which is word 4, then the error correction function enables the correction of the bad word 4 which produces a good word 4 as shown to the right above.

In order for us to do that, we can analyze word 4 with the other words in the sentence. For example, from principle number 3 above, it shows that if a sentence is correct, there must be a similarity between words in that sentence. By examining our incorrect sentence which contains 3 good words and 2 bad words, we can see that there exist similarity between the first 3 good words, but not the last 2 in relationship with the sense of the sentence. Now we can perform the replacement based on that relationship. We can view the replacement process as a replacement of words from a dictionary for example. We can think the incorrect sentence as a set of words, where we look at a dictionary and replace the bad words with the good words. The model below shows the process.



 Copyright 2023 *The Speak Logic Project*

From the model above we replace bad word 4 with a good word from the dictionary and bad word 5 with a good word from the dictionary. This process produces a good sentence which is portable and makes our application execute without error.

The correction process enables us to analyze words in the sentence by using the principle of communication. We already know that if a sentence is correct, there must be similarity between the words in the sentence in relationship with the sense of that sentence. In this case, if a sentence is not correct, that relationship does not hold. From what we have just said, the following principles can be identified.

Principle Number	Principles
1	If a sentence is incorrect, there exits at least one good word in that sentence
2	If a good word exists in an incorrect sentence, it can be used as a basis for correcting the whole sentence.
3	In a given sentence, if a word exists. the entity it points to must also exist. (In other words, in a given sentence, a word does not exist without the existence of the entity it points to)

The way to look at it, we partition the incorrect sentence into two groups, where one is a group of good words and the other one is a group of bad words. In the group of bad word, there exist at least one good word. We pick that good word, and we use it as a basis to perform the correction of the whole sentence. By doing so, we use the good word and we identify the entity that word points to. We analyze that words we also use the principle of communication in relationship with that word. Then from the same set of good words, we identify similar words to correct the overall sentence.

Copyright 2023 *The Speak Logic Project*

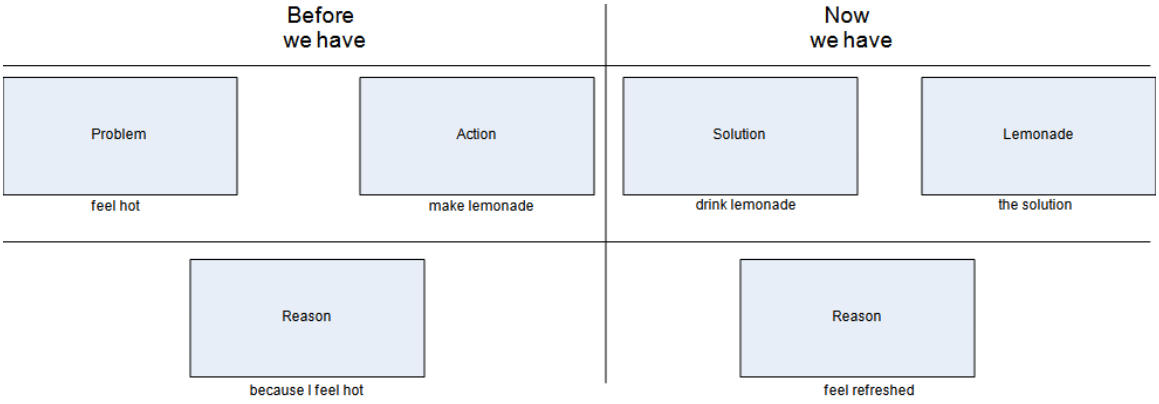
Solution

Now I finish making the lemonade, I drink it in order to feel better. I don't feel hot anymore, therefore that solves my problem. What is important here, before making the lemonade, I felt hot. After making the lemonade, I drink it and feel refreshed. Therefore, the feeling hot problem is solved by the lemonade.

As we have seen from the problem statement, the problem was identified as well as the action taken to solve the problem. After the action, the underlying problem is solved. To better understand that, let's look at it like this.

Before	Now
No Lemonade	Lemonade
Feel Hot	Feel Refreshed

We can also take it like this

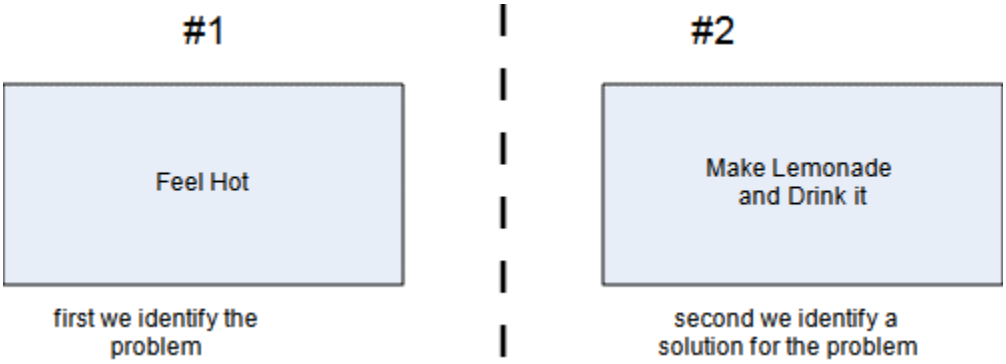


The way to look at it, I did not have the lemonade. I need to make the lemonade in order to refresh myself. After I make the lemonade, I drink it in order to refresh myself. Therefore, by drinking the lemonade, I was able to refresh myself. By drinking the lemonade, I was able to solve my feeling hot problem.

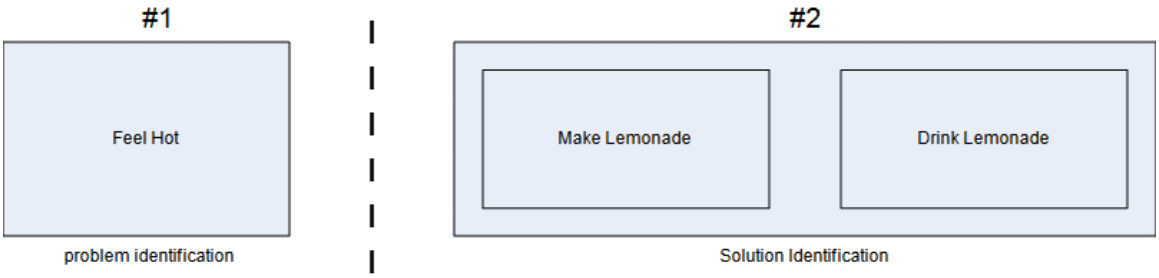
Understanding Problem and Solution

The process of solving a problem enables us to identify the problem first. For this application, the problem was firstly identified, then from identifying the problem, a solution was identified

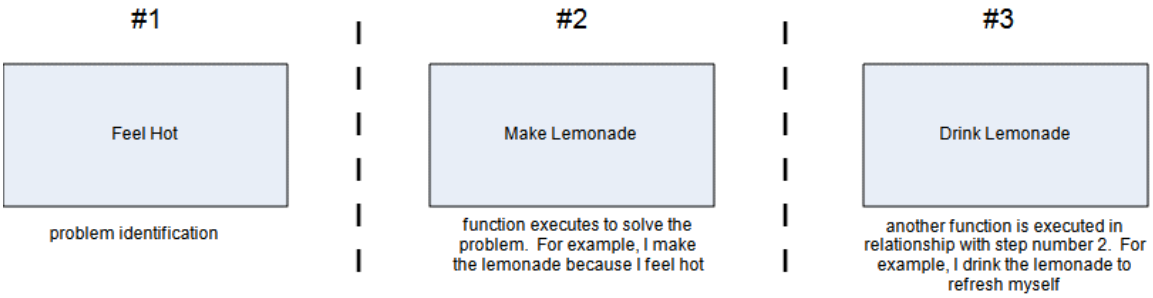
for the problem. To better understand what we have just said, let's take a look of the process again.



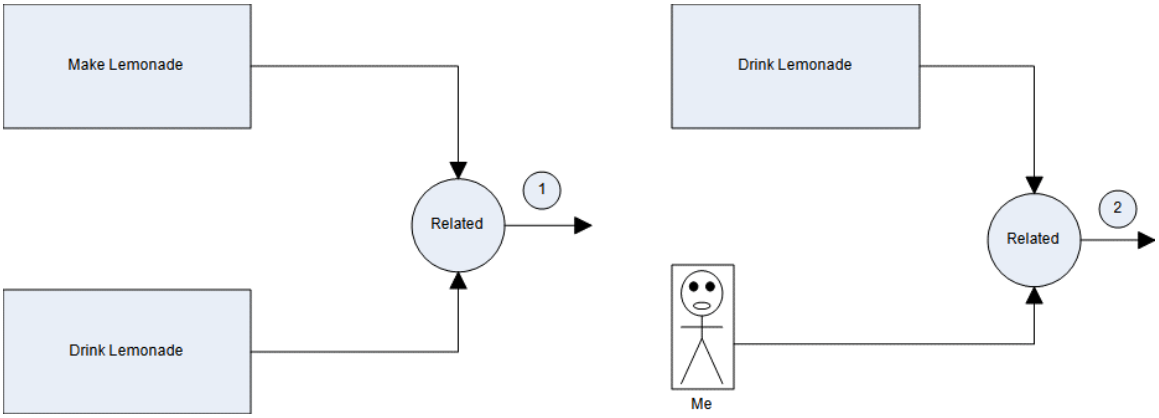
As shown above, first we identify the problem, secondly we identify the solution for the problem. The second step can be viewed as one step, it can also be viewed as two steps. For example, by making the lemonade and drinking it, I feel refreshed, therefore that solves the problem. In another approach, make the lemonade and drink the lemonade can be viewed as two separate entities. Where make the lemonade can be viewed as number 2 and drink the lemonade can be viewed as number 3.



As shown above, first we identify the problem, which is viewed as the problem identification process. Secondly we identify the solution for the problem, which is viewed as the problem solution identification process. While making the lemonade function is a part of solving the problem, however making the lemonade without drinking it does not solve the problem. What is important here, the solution of the problem must be related or connected to the entity that has the problem. The way to look at it, I have the feeling hot problem, I must drink the lemonade that I make in order to feel refreshed to solve that problem. In this case, we can see that we have 3 entities here: we identify the problem which is viewed as the first entity; secondly we execute a function to solve that problem which is viewed as a second entity; third, a function is executed in relationship with the function that tends to solve the problem or in relationship with entity that is produced to solve the problem. Let's show that by the model below.



To better understand the process, we can relate step number 2 with step number 3. For example, make the lemonade is related to drink the lemonade.



As shown by the model above, the solution process enables the two functions to the left to be related. If they are not related, then that solution does not exist. To the left at node 1, we can say that in order for the problem to be solved, the lemonade must be drunk by me. To the right, we can see that there is a relationship between me and the drinking of the lemonade. That further verifies the model to the left. That relationship enables me to drink the lemonade in order to solve the problem. If there is no relationship or that relationship cannot be verified, then the identified entity which is claimed as the solution for the problem is not a solution for that problem at all.

Conclusion and Summary

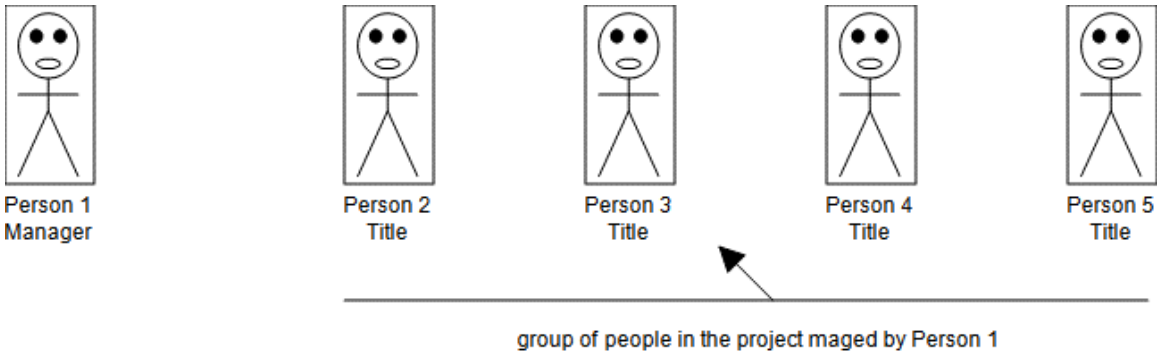
In order to solve a problem, we must first identify the problem. After we identify the problem, we can identify a solution for the problem. After we identify the solution for the problem, that solution must be related to the entity that is affected by the problem. If there is no relationship between the underlying solution, then that entity is not considered to be a solution for that problem. In other words, the function that we executed to solve the problem or the entity that we

 Copyright 2023 *The Speak Logic Project*

produce to solve the problem must be related to the entity that has the problem. The entity that has the problem must execute a function related to the underlying solution to solve the problem.

Understanding External Function

In the communication domain, our application is a function of communication. In terms of function management, in the communication domain a manager uses communication to manage the function of employees in the application. To better understand the process, let's take it like this.

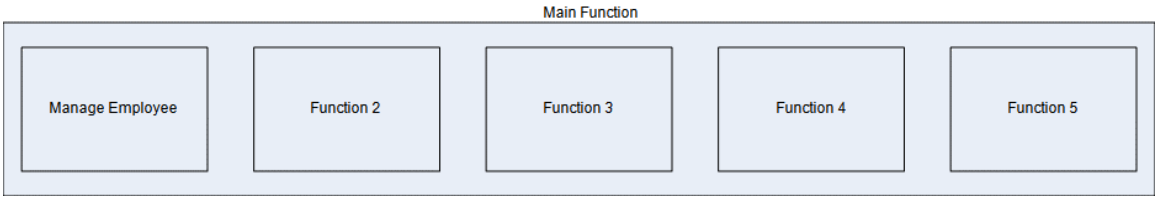


The model above shows a group of people in the project managed by Person 1 who is the manager of the project. The group of people also includes Person 1 as well. The fact that the function of the project is a function of communication and the manager manages the function of the project. The fact that the function of the project is a function of communication and the function of the project includes the function of everybody in the project, the manager manages the function of everybody in the project. To better understand that, let's take it like this.

Person in Project	Function
Person 1	Function 1 or Manage Employee
Person 2	Function 2
Person 3	Function 3
Person 4	Function 4
Person 5	Function 5

In order for the application to execute, each function must be combined to form the main function of the application. From what we have just said, we can show the main function of the project in the form below..

Copyright 2023 *The Speak Logic Project*



From the model above, *Manage Employee* is the function of the manager where each employee in the application has his/her function. Now, since the main function is a function of communication, each function in the project is a function of communication as well. For example, *Manage Employee* which is the function of the manager is a function of communication. As well as *Function 2*, *Function 3, Function 4* and so forth are functions of communication.

Given that the people in the project depend on the principle of communication to execute those functions and one cannot apply the principle for each other, therefore only people in the project can execute functions of the project. In other words, *Manage Employee* is the function of the manager and that function is a function of communication of the manager, only the manager of the project who is Person 1 can execute that function. Another person cannot execute that function for the manager. Again, *Manage Employee* is the function of the manager in the project and another person outside the project or organization cannot assume or take or execute that function for the manager. *Function 2* of Person 2 in the project can only be executed by Person 2 in the project, another person cannot execute that function for Person. The way to look at it, a person outside the project or application cannot be a part of that application to execute a function for people in the application.

To better understand that, let's look at like this. At the beginning of the project, *Manager 1* assigns *Function 2* to Person 2 in the project. Where *Manager 1* may provide a description of that function to Person 2. In this case, the execution of that function is a function of communication of Person 2 in the project. Another person who is not in the project or either outside cannot come and execute that function for Person 2, because that function is a function of communication of Person 2. That person cannot communicate for Person 2; therefore that person cannot execute that function for person 2.

We cannot apply the principle for each other to execute functions for each other in an application; it is important for us to understand functions of people in our application and not to assume functions of other people. Given that one cannot apply the principle of communication to communicate for each other, it is not possible for one to assume each other function in a

 Copyright 2023 *The Speak Logic Project*

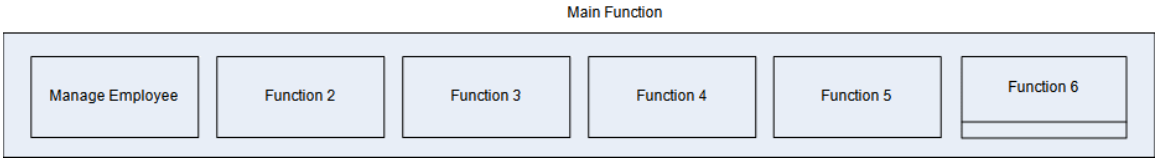
project. For this reason, if our project requires the execution of a function, it is not possible for us to assume that function. In this case, it is better to enable a person or people outside execute the function for us.

To better understand the explanation above, let's look at it like this. Since in this application we cannot assume or take other people functions, our application may require another function in order to execute. In this case, we can allow other people outside of our organization to execute that function. For example, in this application the function of the application is executed by people in *Organization 1*. If another function is required to execute that function, it may not be possible for us to assume that function. In this case, we can allow a person from another organization to execute it. Let's show that by the table below.

Person in Project	Function	Organization
Person 1	Function 1 or Manage Employee	Organization 1
Person 2	Function 2	Organization 1
Person 3	Function 3	Organization 1
Person 4	Function 4	Organization 1
Person 5	Function 5	Organization 1
Person 6	Function 6	Organization 2

From the table above, *Function 6* is a function that is required in order to execute the project. Since *Function 6* cannot be executed by people in organization 1, Person 6 from another organization executes *Function 6* in Organization 1. The way to look at it, people in the project recognize *Function 6* and cannot assume it, therefore Person 6 can comes to organization 1 to execute function 6.

From the explanation above, *Function 6* can be viewed as a subfunction or external function. In this case, *Function 6* can be viewed as a subfunction of the main function as shown below.



From the model above, *Function 6* is viewed as an external function. *Function 6* is not managed or may not be managed by the manager who manages the employee. In some cases, depending on the function, it may be a function that is not assigned by Person 1 who is the manager and it is not described by Person 1 also. Depending on the function, in term of description, *Function 6*

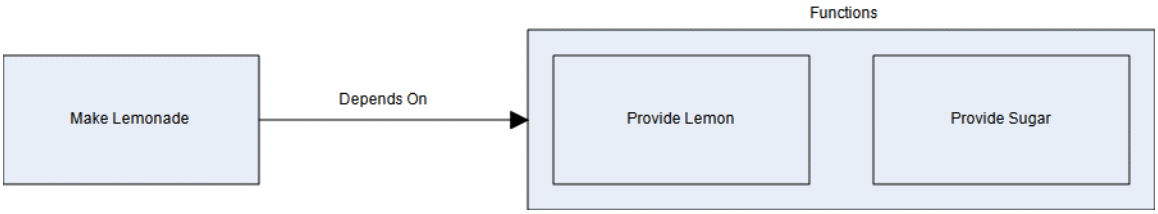
 Copyright 2023 *The Speak Logic Project*

may not appear in the model. In other words, *Function 6* may appear as *Function 6* in the model without a description. Depending on the function, sometimes it may take a better understanding of the principle. What is important here is the fact that one cannot apply the principle for each other, in our project we cannot assume or take other people's functions. Even in our application, we must allow other people to execute their functions, even when our application needs those functions to execute.

In summary, there are two ways to understand this section. First, given that the function of our application is a function of commination and one cannot apply the principle to commutate for each other, another person or a person outside cannot come to the project and execute a function for a person in the project. Second, given that our application is a function of communication and one cannot apply the principle to communicate for each other, it is not possible for one to assume each other's functions. Because of that, we have to allow other people to execute functions that are needed in our application where those function are not assigned to us and cannot be executed by us.

Understanding Function Dependency

A function may depend on another function to execute. A function may depend on another function to execute properly. There are two ways of looking at functions dependencies. A function dependency can be viewed as internal; a function dependency can also be viewed as external. For example, in this application the making of the lemonade function depends on the providing lemon function and the providing sugar function. Here, let's show that by the model below.

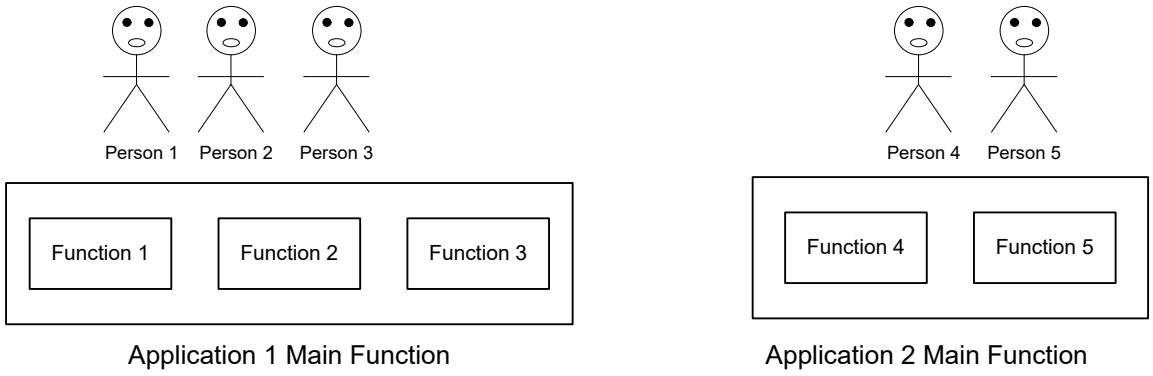


From the model above, since the *Make Lemonade* function is a function of communication and *Provide Lemon* and *Provide Sugar* are also functions of communication, therefore the making of the lemonade function depends on the communications of both Ms. Smith and Peter. The way to look at it, the providing lemon function is a function of communication of Ms. Smith and the providing of sugar function is a function of communication of Peter, thus the making of lemonade function depends on communications of Ms. Smith and Peter. Since both Ms. Smith

and Peter are in the application, in this case we call this function dependency process as internal function dependency.

Now, let's take a look at external function dependency. An external function dependency is viewed as a function that is not in our model, however in order to solve the problem our application intended to solve properly, that function must be executed. For example, in our application we have 3 functions. Our application is modeling with *Function 1, Function 2*, and *Function 3* where our application when it is executed it must solve *Problem 1*. In our application, we have *Person 1*, *Person 2*, and *Person 3* where we have just listed the function of each person previously. In order for the function of our application to execute to solve *Problem 1* properly, another function which we may call external function must be executed.

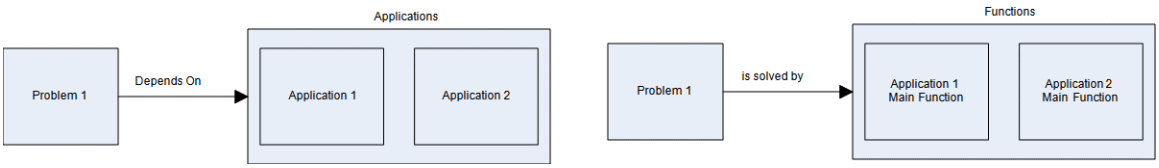
From the explanation above, the external function is a function of communication. We can call it *Function 4*. In this case, *Function 4* is a function of communication of people who are not in our application. To better understand that, let's show that by this model.



From the model above, *Application 1* is made of *Function 1*, *Function 2*, and *Function 3,* where *Application 2* main function is made of *Function 4* and *Function 5*. In order for Application 1 to execute to solve Problem 1 properly, Application 2 must be executed properly. Here, the main function of *Application 2* is viewed as an external function that Application 1 depends on. That function is a function of communication of *Person 4* and *Person 5*. What is important here is the function of Application 1 depends on the function of Application 2, however Application 2 is not in the model. The people in Application 2 are not in Application 1.

Given that Application 1 needs Application 2 in order to execute properly to solve the problem it intended to, with the understanding of the principle, it may be possible to provide feedback to people in Application 2 to enable the execution of Application 2 properly in order to solve *Problem 1* which needs to be solved by Application 1 when it is executed. Since Application 1

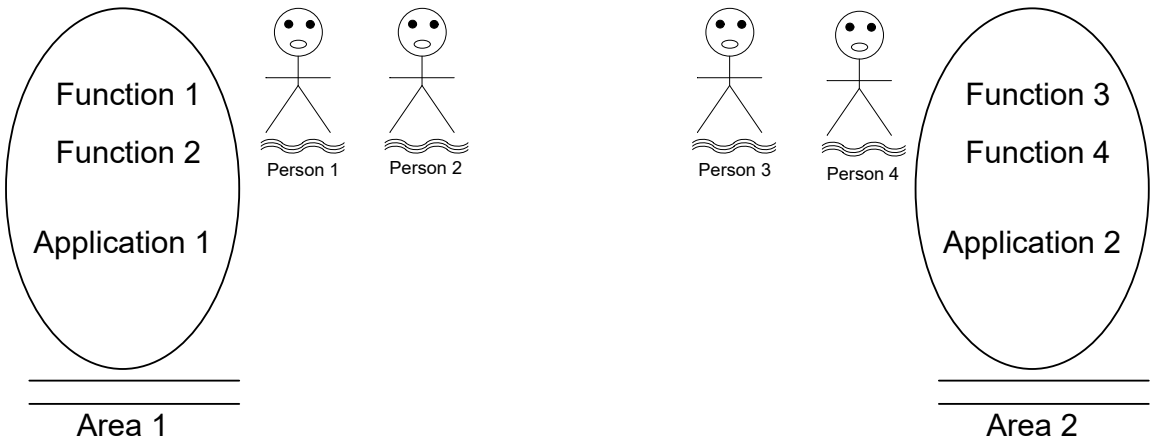
needs to be executed with the help of Application 2 to solve Problem 1, in this case the execution of Application 2 also solves Problem 1. From that, we can see that problem 1 is solved by both Application 1 and Application 2 as shown by the model below.



In summary, the people in one application cannot be present in the other application in order to execute function of that application to make it executes properly; however with the understanding of the principle, it may be possible to provide feedback to people in one application in order to enable the execution of the other application to solve the underlying problem.

Understanding our Functions Related to Locations

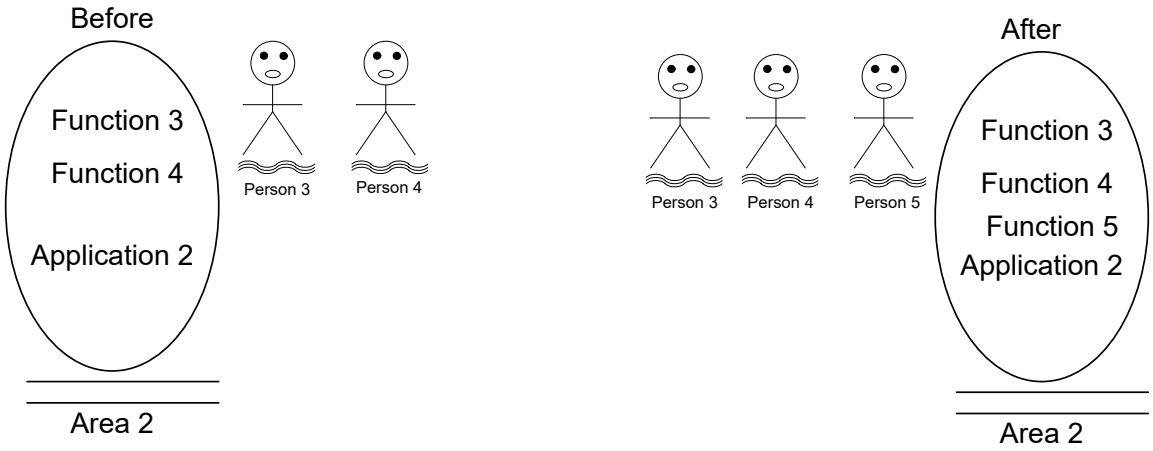
The function of our application is a function of communication. The function of our application is executed in an area. That area where the function is executed is considered to be our working area. The area where the function of the application is executed, is considered to be the working area of the people in that application. For example, in this application, the overall function of the application is executed at my location, where the providing lemon for making the lemonade is executed at Ms. Smith's location and the providing sugar for the lemonade is executed at Peter's location. The location where the application is executed is considered to be fixed, whether the people in the application are considered to be mobile. In this case, the people in the application can move from location to location, however the location itself cannot be moved. To better understand the overall explanation, let's take a look of the model below.



Copyright 2023 *The Speak Logic Project*

From the model above to the left, we have *Application 1*, where *Application 1* is made of *Function 1* and *Function 2*. Application 1 is executed by *Person 1* and *Person 2* in *Area 1*, where Function 1 is the function of Person 1 and Function 2 is the function of Person 2. To the right, we have *Application 2*, where Application 2 is made of *Function 3* and *Function 4* and executed in *Area 2*. Application 2 is executed by *Person 3* and *Person 4*, where Function 3 is the function of Person 3 and Function 4 is the function of Person 4.

Because of the portability of the principle and the mobility of the people in each area as well as the relationship between people in the application—we mean the relationship of the person entity—there may be a problem in Area 2 that needs to be solved, where people in Area 1 may help execute a function to solve that problem. With the understanding of the principle, it is possible. In this case, another application can be created to add a person from Area 1 to the application in Area 2 in order to solve that problem. Another way to look at it, in Application 2, another function can be added to that application by updating that application to include a function for a person from Area 1, where the function of that person in Application 2 in Area 2 is to solve that problem. The way to look at it, the understanding of the principle enables a person from an area to be part of another application in another area in order to solve that problem. To better understand that, let's take a look of the model below.



Understanding our Function in our Application

We have learned previously that the function of our application is a function of communication and each person in our application applies the principle of communication to execute our application. From what we have just said and from what we have learned before, the function of each person in our application is a function of communication. For this application, the making

 Copyright 2023 *The Speak Logic Project*

of the lemonade is a function of communication so does the providing lemon and sugar for the lemonade are functions of communication. For example, the making of the lemonade is a function of communication which includes my communication, Ms. Smith's communication, and Peter's communication.

By applying the principle of communication, we can identified and correct errors in the communication to enable our application to execute without error. From that, we can see that the function of our application which is a function of communication is verified or validated by the principle of communication. What is important here? Given that the function of our application is a function of communication and the communication of our application is validated by the principle, therefore the function of our application is also validated or verified by the principle of communication.

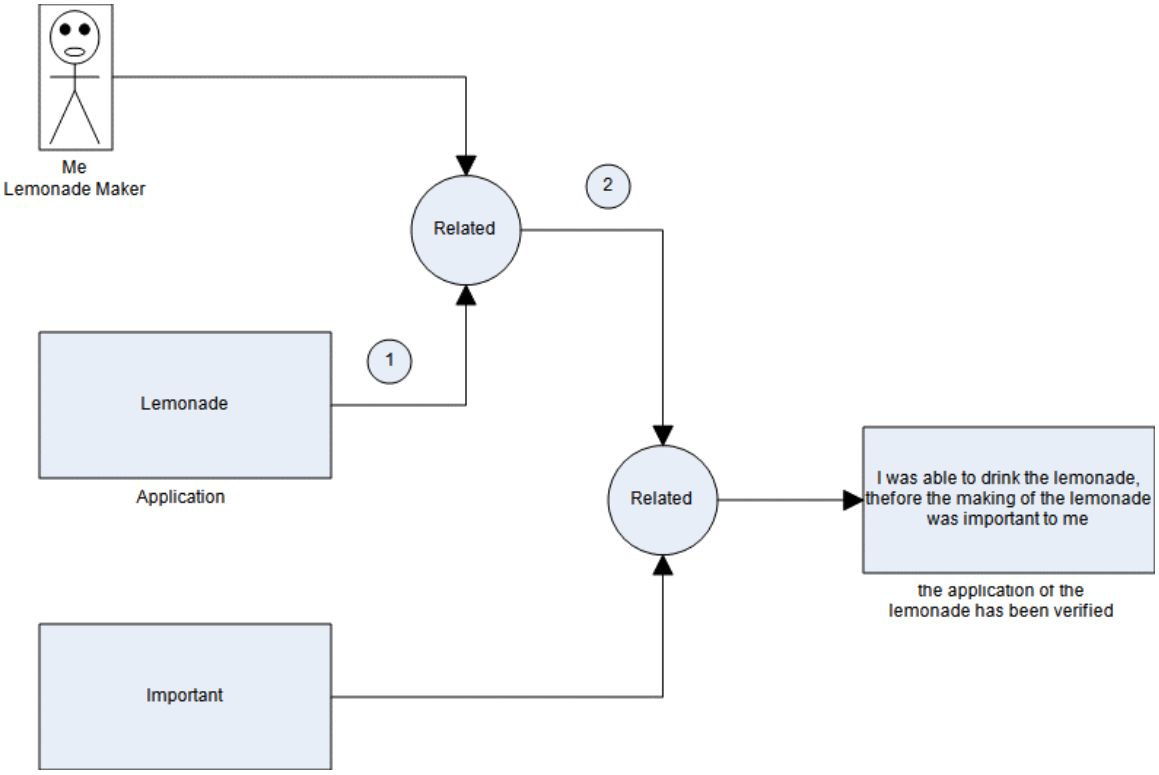
For example in this application, my function which is *Make Lemonade* is validated by the principle of communication, so do M.s Smith's function *Provide Lemon* and Peter's function *Provide Sugar* are also validated by the principle of communication. For this reason, in our application we can identify the function of each person in the application and verify that each function is indeed valid. For example, in this application, the making of the lemonade function is a valid function, because it is validated or verified by the principle of communication. The providing lemon for the lemonade function which is the function of Ms. Smith is also a valid function, because it is validated by the principle of communication.

Overall, in our application is always good for us to identify our function or the function of each person in the application and show that the function of each person in the application is indeed a valid function. Another way to look at it, given that everybody in the application applies the principle to execute the function of the application, in this case the function of the application is indeed validated by the principle. We can take it like this. The function of the application is a contribution of the function of everybody in the application, since the overall function of the application is validated by the principle, the function of everybody in the application must be validated by the principle as well. We can also look at it the other way around, since the function of everybody in the application is validated by the principle and the overall function of the application is a contribution of everybody in the application, the overall function of the application is also validated by the principle.

Conclusion and Application Verification

 Copyright 2023 *The Speak Logic Project*

Now that I finished making the lemonade, I drink it and I feel refreshed. By drinking the lemonade, my feeling hot problem has been solved. Drinking the lemonade and feeling refreshed, shows that the lemonade application is important since it enables me to solve the feeling hot problem. Here, we can verify that using the principle entity.

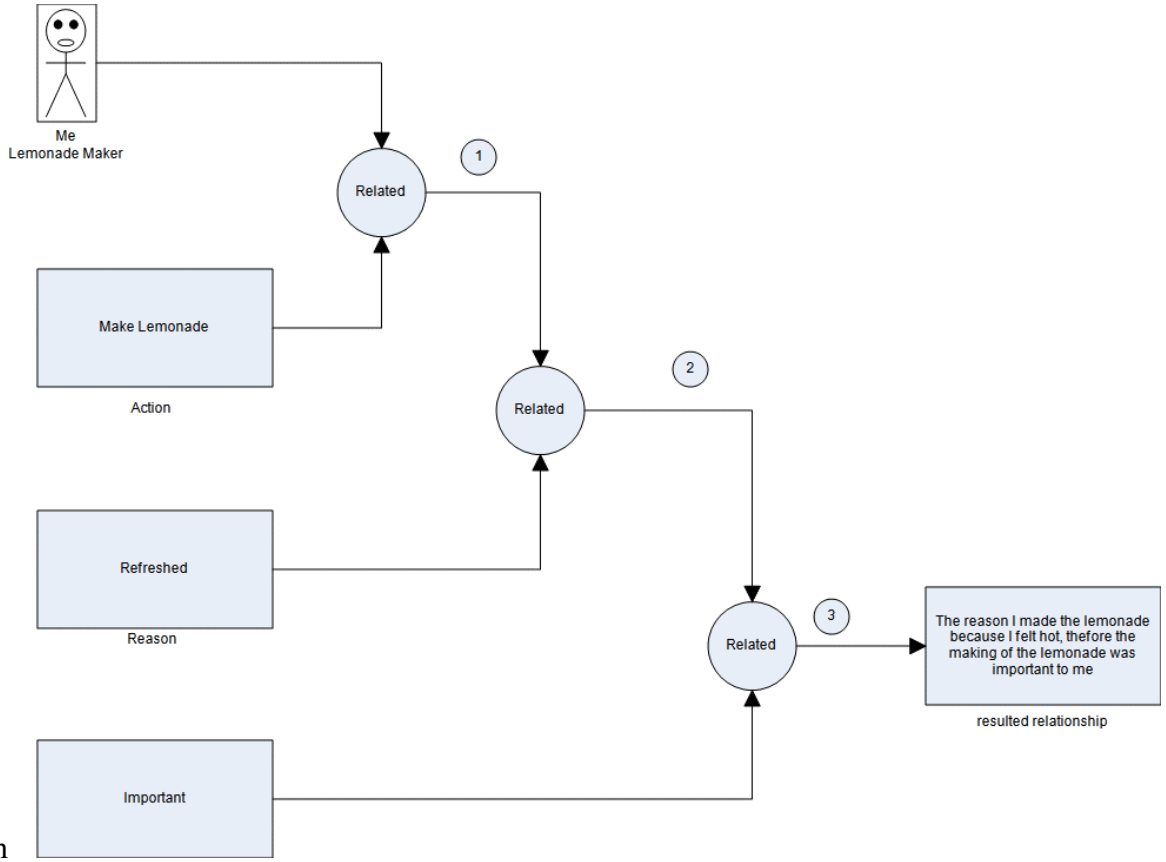


As shown above, the way to look at it, the lemonade application is important to me because I was able to drink it to solve the problem. As we can see, we use the principle entity in relationship with the application to verify that application. For example, at node 2 it shows that *I am able to drink the lemonade*. Given that we are related with the principle, we are also related to each aspect of the principle. In this case, if our application is correct, we can use the principle entity to verify that. We can also think the end relationship above like this. *I was able to drink the lemonade, therefore the lemonade was important to me.*

All what we have done here was verifying our function or our application. We can also say that the relationship above enables us to verify the existence of our function or the existence of our application. In other words, that relationship enables the creation of the project. That relationship enables us to create a project to make the lemonade.

 Copyright 2023 *The Speak Logic Project*

In relationship with the problem statement, we can see that our function is verified even before completed the project. Let's show the process here. *Make Lemonade* was our action and *Feel Refreshed* was our reason. We can put that in this form



As we can see the application can be verified with the problem statement. For example, before making the project or starting the application, we can analyze the problem statement and verify the application. As shown above, as an example, at node 1 the action of the problem statement can be analyzed in relationship with each person in the application. In this application, the making of the lemonade action can be analyzed by Me, Ms. Smith, and Peter or by taking Me, Ms. Smith, and Peter into consideration. The result of the relationship can also be analyzed by taking Me, Ms. Smith, and Peter into consideration in relationship with the reason of the application. This process is being shown at node 2. Finally, the principle entity can be used to verify the result of that relationship. As shown above, at node 3, the verification is being done in relationship with Me, Ms. Smith, and Peter.

From the model above, we can see that before Application 2 in Area 2 includes two people with two functions. Then another person from Area 1 is added to the application in Area 2, where that

 Copyright 2023 *The Speak Logic Project*

person is Person 5 in area 2 and the function of that person is Function 5. In this case, the manager of that application add a function for Person 5 in the application where both the manager does have a problem statement for that function as well as Person 5 does have a problem statement also. In this case, when Application 2 is executed in Area 2 with the inclusion of Function 5, then that will solve the underlying problem.

Copyright 2023 *The Speak Logic Project*

Understanding Complexity in the Communication Domain

Introduction

Complexity is viewed as increasing relationships where it takes more effort for us to understand. We can define a noncomplex entity as an entity that has few relationships while a complex entity is one with many more relationships.

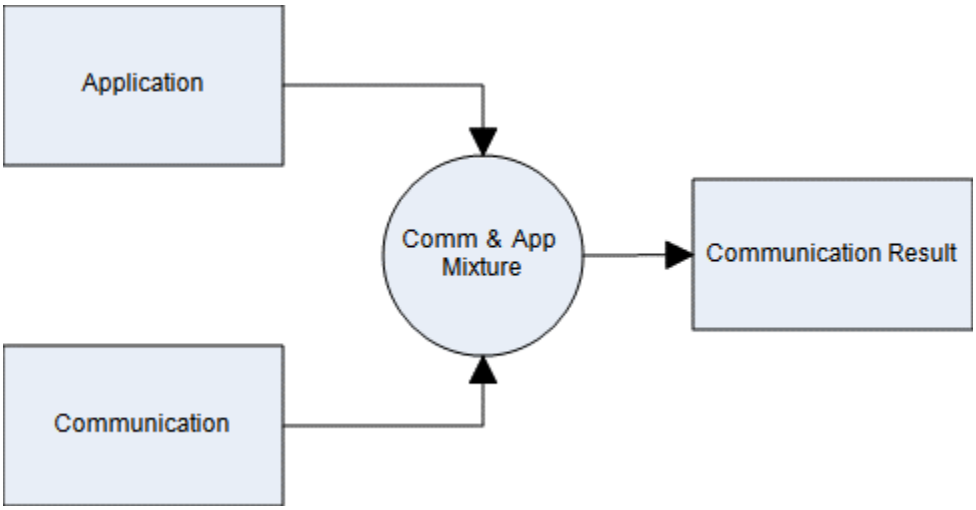
In the communication domain, our application is a function of our communication. Since our application is communication driven, it depends on our communication. In terms of dependency of our application from our communication, it is always good to look at the size of our communication. For example, less communication in our application renders our application less complex than an increase of communication in our application. What do we mean by increased communication in our application? Since our application depends on our communication, we can say that the increase of unnecessary of communication in our application. Since those communication also drives our application, they render our application to be more complex even when they are not needed to execute the function of our application.

In this chapter, we are going to look at the effect of complexity in our application in the communication domain. We are going to look at the communication that drives our application and identify complexity that enables our application to perform less. We are going to show the process of reducing complexity which enables our application to perform much better.

Copyright 2023 *The Speak Logic Project*

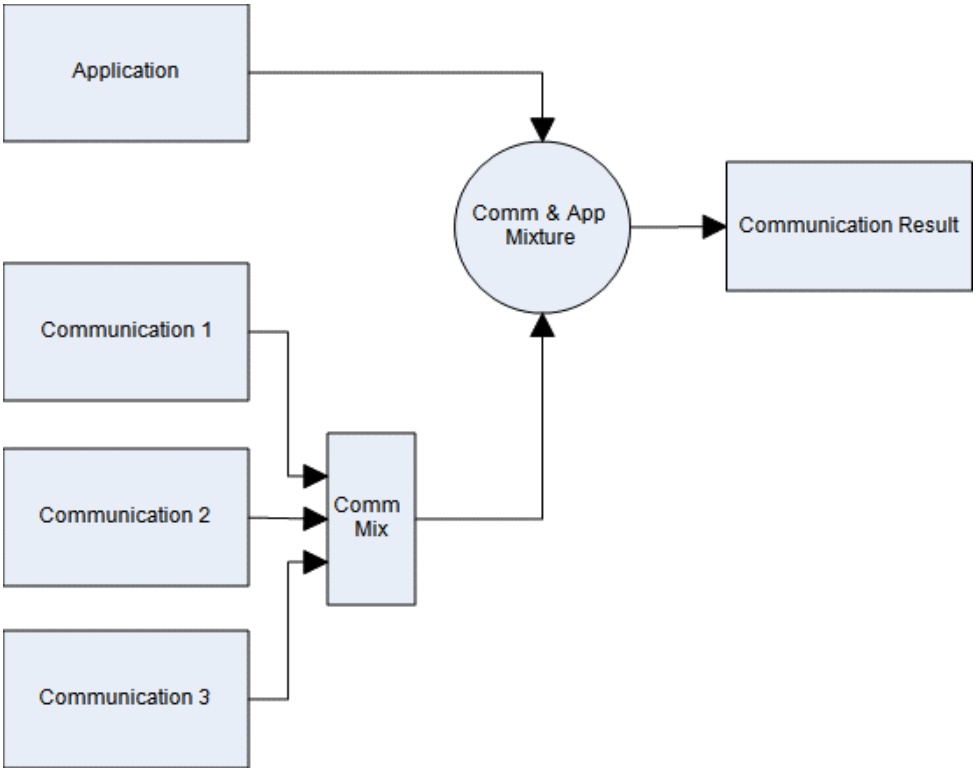
Complexity in Communication Domain

In the communication domain, the function of our application is given in the form below:



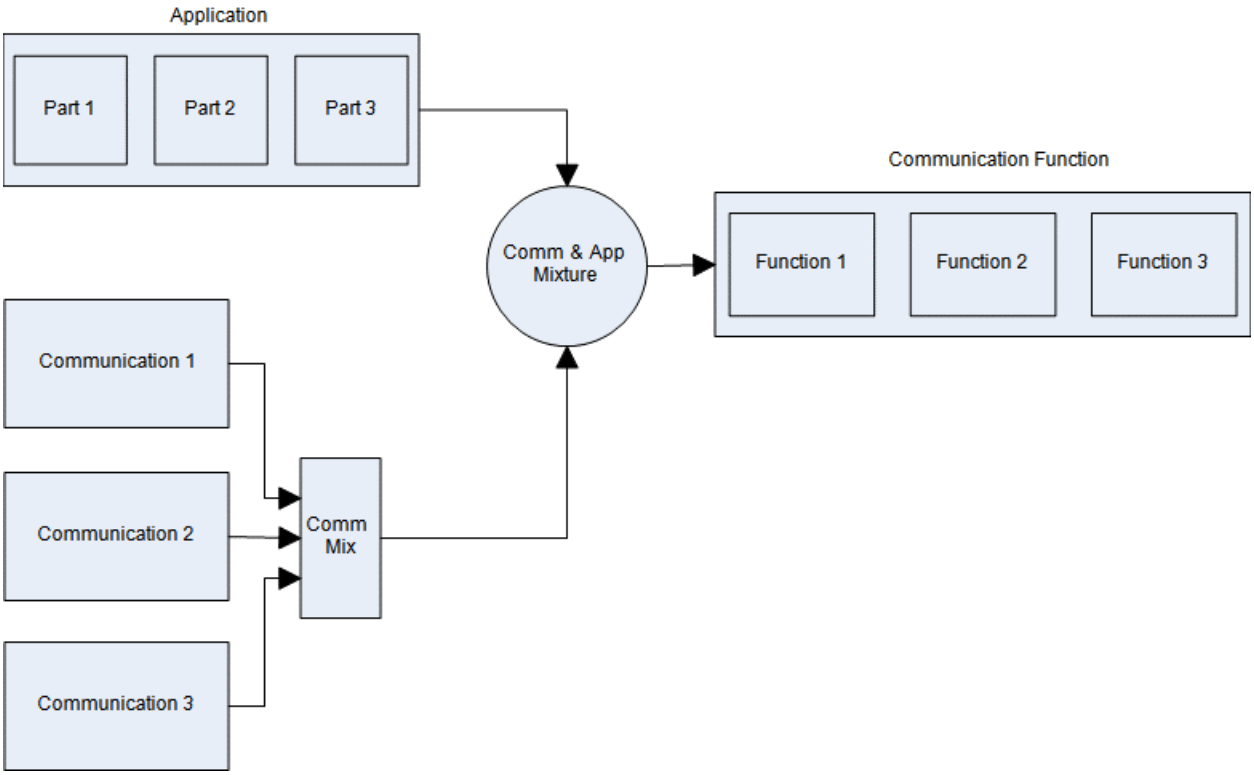
From the model above, we can see that the identified application is driven by that communication. In this case, we can say that only communication that enables that application drives that application. The model above shows a typical model, only one application is shown. Here, let's show a more expandable model with more communication that drives that application.

Copyright 2023 *The Speak Logic Project*



From the model above, our application includes multiple communications. In this case, we can say that our application includes communication from multiple people. Since our application includes multiple people, the idea of our application must include ideas from multiple people as well. In this case, we can say that the idea of the application is partitioned to multiple people. We can put that in the form below which includes the function related to each communication.

 Copyright 2023 *The Speak Logic Project*



From the model above, our application is driven by multiple communication in relationship with multiple ideas. For example, Communication 1 is from a person in the application where that person has Part 1 of the idea of the application; so do Communication 2 and part 2 of the idea of the application. From above, we show that idea of the application as Part 1, Part 2, and Part 3 where each idea is from a separate person. To the right, we show the function of each person. For example, Function 1 is the function of Person 1 in relationship with Part 1 which is that person's idea and his/her communication which is communication 1.

From the model above, we can see that the application is driven only by the identified communication. In this case, we can say that Function 1 is driven by Communication 1 in relationship with Part 1 of the idea of the application. For instance, if Function 1 is the function of Person 1, we can say that Function 1 is driven by Communication 1 of Person 1 in relationship with Part 1 which is the idea of that person. The same process happens for Function 2 and Function 3 which are driven by Communication 2 and Communication 3 in relationship with ideas Part 2 and Part 3.

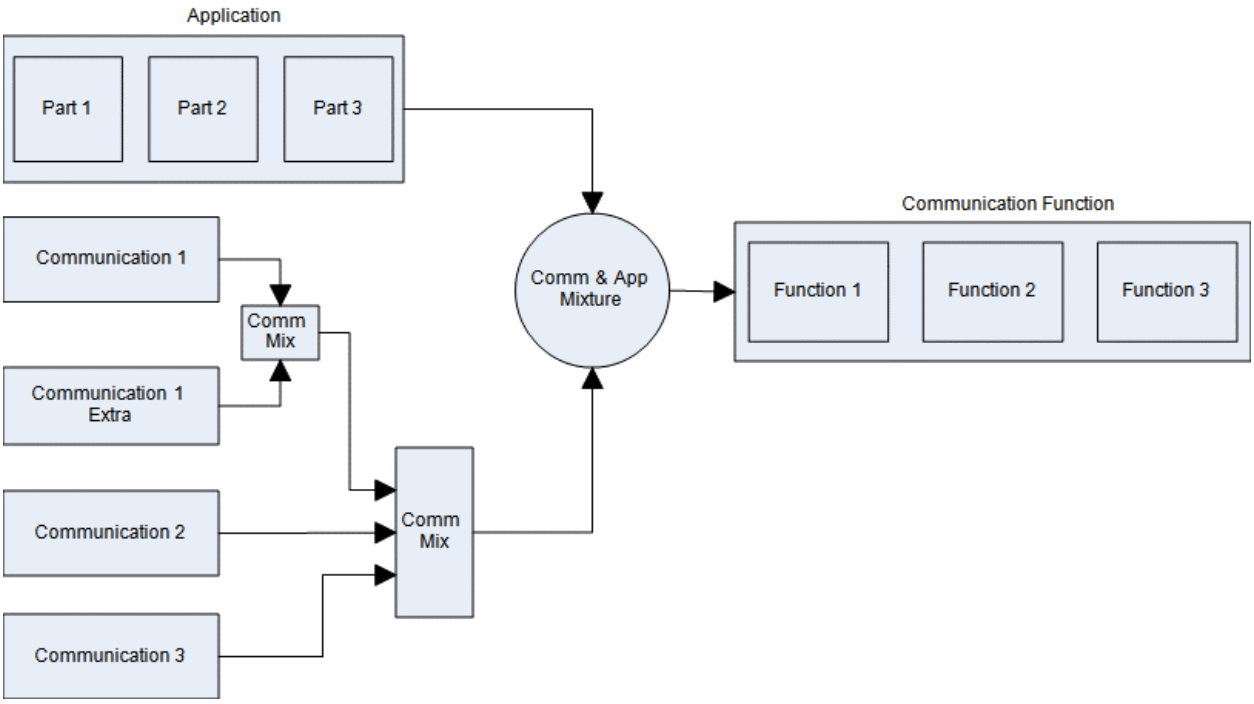
What we have learned from the model above is that only the communication the application depends on drives that application. We can say naturally that application depends on its own communication. Only the natural communication that application depends on enables to drive

 Copyright 2023 *The Speak Logic Project*

that application. What we mean by natural communication for that application? We mean only communication that enables that application without any extra.

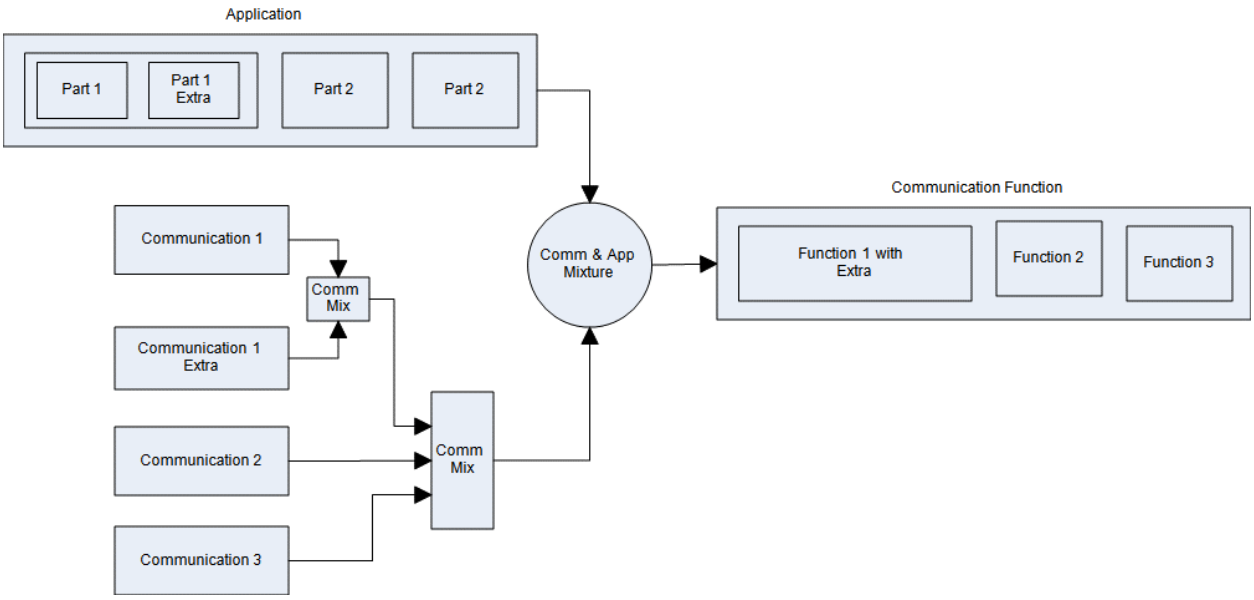
Introduction of Complexity in our Application

In the communication domain, extra communications that do not enable the application add complexity to that application. We call those extra communications because they are not considered to be natural communication for that application. They are considered to be added to the natural communication that enabled the application. To better understand the process, let's take it like this by representing the model below.



From the model above, Communication 1 Extra is viewed as extra communication that is added to the communication for that application, and it is not natural for that application. Visually, we can see the model becomes more complex comparing to the previous model. We show the model only from above, given that there is a relationship between the communication and idea of the application; and the function is executed accordingly. In this case, it makes sense to show the extra of the idea of the application as well as the result of that application. Let's take a look at the model below as an expandable view.

Copyright 2023 *The Speak Logic Project*

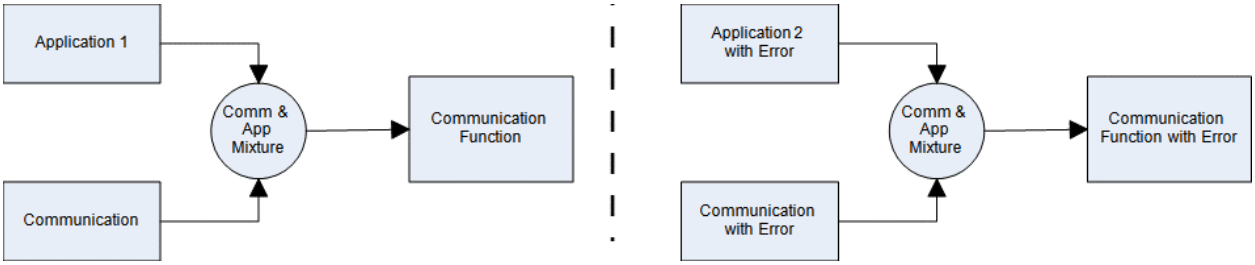


The model above shows that the extra communication added to the application enables extra ideas to be added to the application. They enable the function of that application to be executed in relationship with the extra communication and the extra idea. In this case, we put Function 1 with extra to show that the function of the application is driven with other communication which is not natural to that application. If we were going to expand that model in a simpler form to show more extra added to the application, this will result in a more complex form of the model above. All that we need to know here is that in the communication domain, extra communication added to our application enables extra ideas to be added to the application which results in the application being complex. In this case, that application is no longer driven by its natural communication.

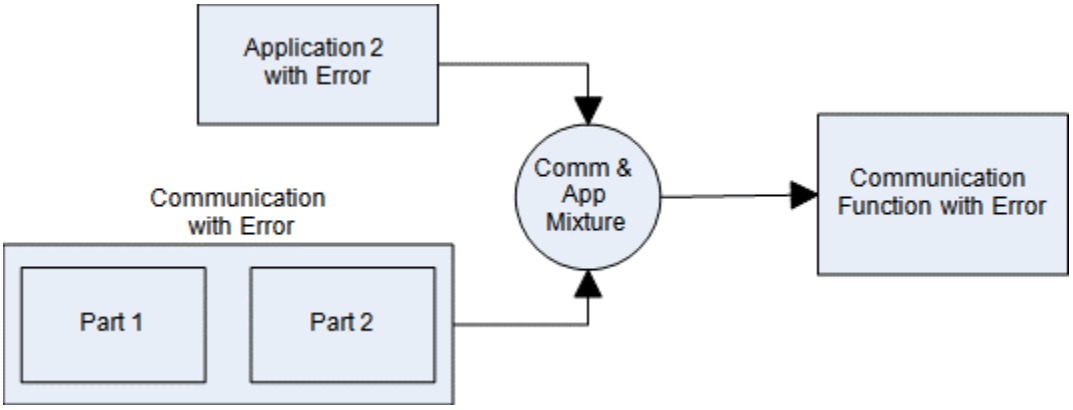
Error in Communication in Relationship with Complexity

Our application is driven by communication; any error in our communication enables errors in our application. This also adds complexity to the application. The way to look at it, introduction of errors in our application enables our application not to be driven by its natural communication. In this case, those errors also drive the application. To better understand the overall process, let's show two applications side by side where one is driven by its natural communication where the other one includes error in communication.

 Copyright 2023 *The Speak Logic Project*



From the model above, we can see that the application or the function to the left is driven by its natural communication. From the one to the right, that application is not driven by its natural communication. Rather, we can see that the application to the right is driven by a communication that contains errors, which enable errors in the resulted function. From the model to the right, if we want to, we can represent the communication into two parts: one part that is normal and another part that is viewed as error in the form below.



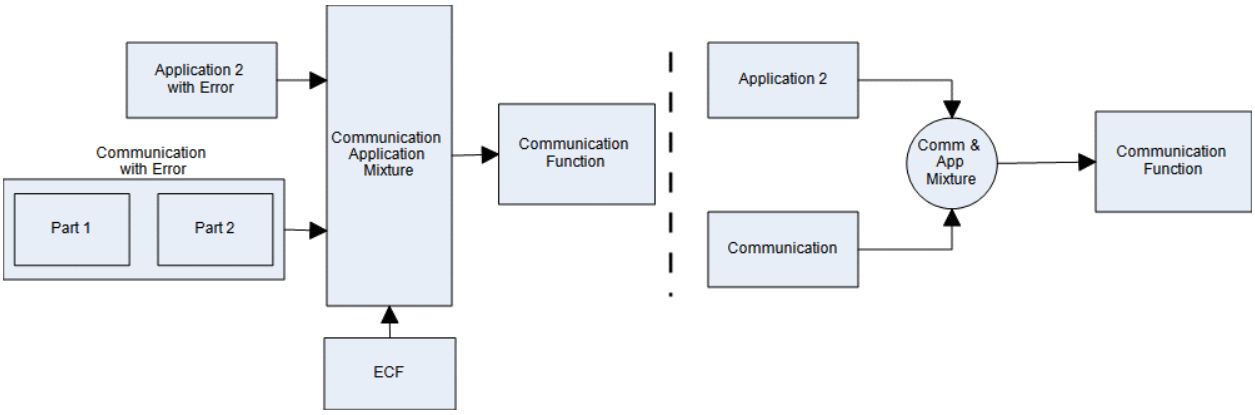
From the model above, we can think of communication as an actual sentence. In that sentence, we have some words that are good and some words that are bad. The good words can be viewed as part 1 which is the good part of the communication. The bad words are viewed as part 2 which are simply errors. Clearly, by looking at the two models above, we can clearly see that the model of the application driven by its natural communication is less complex than the second model. If we want to expand the second model to include more communication as we have done previously, we would see that it is much more complex compared to the natural form.

Error Correction Reducing Complexity

From the model above, we can see that the application is no longer driven by its natural communication. From the feedback process, we know that the Error Correction Function (ECF) enables us to correct errors in our application so it can execute properly. In this case, we can

 Copyright 2023 *The Speak Logic Project*

simply introduce the correction process to the model. By doing so, we have the model below.



From the model above, to the left we add the error correction process to the communication in order to remove errors in the communication. To the right, we show another model. The model to the right shows a resulted model with error removal. We can say that, as a result of removing errors in the communication, a less complex model of the application is produced . Clearly, we can see that the model to the right is less complex than the one to the left. We can call the model to the right the natural form of our application. We can also say that application is driven by its natural communication.

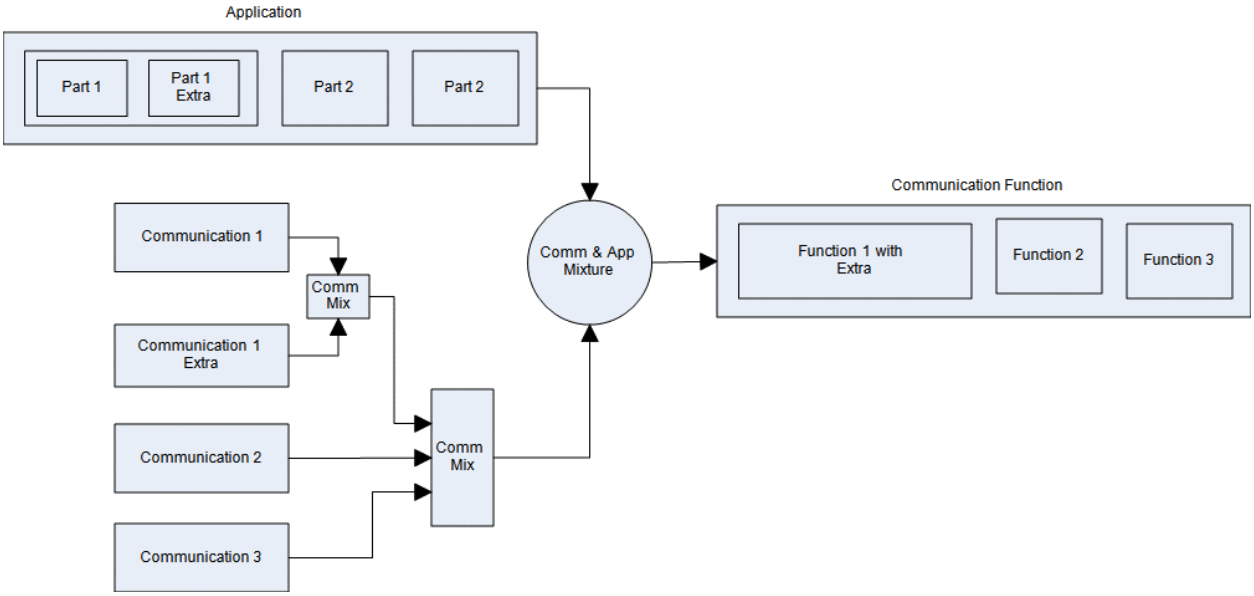
To better understand the overall process, we can think it like this. While we are communicating about our application, our parent heard us and provided us with feedback which enables us to make the correction to our communication. From that we can see two cases: if we did not receive feedback from our parent, then our communication is correct. That communication is viewed as a natural form of communication for that application. We can also say that our communication is naturally correct because it is disregarded by our parent in terms of correction. On the other hand, if we receive feedback from our parent, then our communication is not correct. Therefore, that feedback is provided to us in order to make correction in that communication so it can be in its natural form. From that, we can see that natural form is what drives our application to remove complexity in it.

Complexity and Idea of Application

We already know that there is a relationship between the idea of our application and our communication. Our communication affects the idea of our application. If our communication contains errors, then that renders the idea of our application not to be good. In terms of complexity, introduction of unwanted communication in our application enables unwanted ideas

 Copyright 2023 *The Speak Logic Project*

to be introduced in our application. To better understand this process, let's look at the model below.

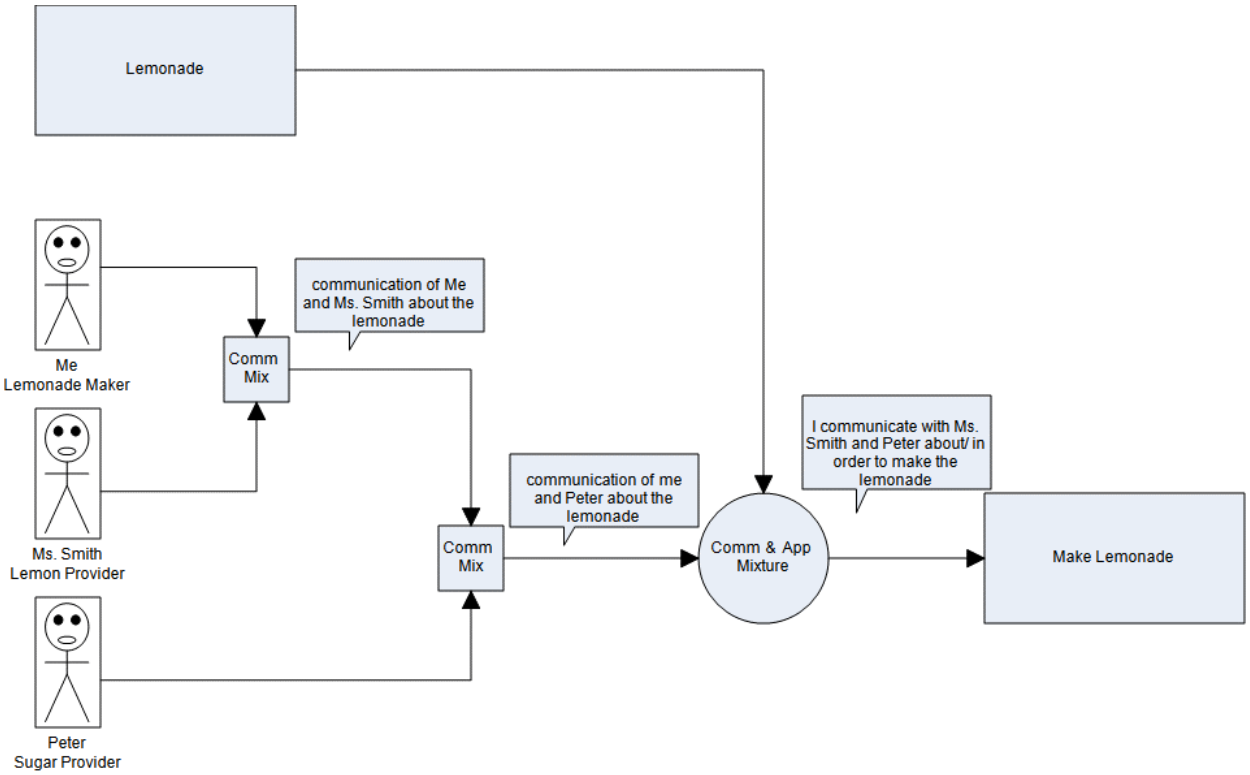


From the model above, the extra communication added to the application is viewed as unwanted communication because it is not needed to execute the application naturally. From the same model, Part 1 Extra is viewed as an extra idea added to the application that is not wanted by the application. As we can see, the unwanted or extra idea added to the application enables the application to become more complex. What is important here is the relationship between the communication and the idea of the application that enables whenever extra or unwanted communication is added to the application, unwanted or extra ideas also added to the application. While we show that in the form above to reduce space, the more extra added the more complex the application is and the model above can be expanded much more.

Complexity and People in the Application and their Functions

Since the application is a function of communication and the communication is from people in the application, those people are responsible for the complexity in the application. In this case, the communication of people in the application can cause the application to become more complex. It can also cause the application to become less complex. To better understand the process of reducing complexity in the application in terms of communication, it is always good to think that in our application, only communication that enables is part of that application. Communication that cannot enable that application is not part of that application. To better understand the explanation, let's take a look at the lemonade application.

 Copyright 2023 *The Speak Logic Project*



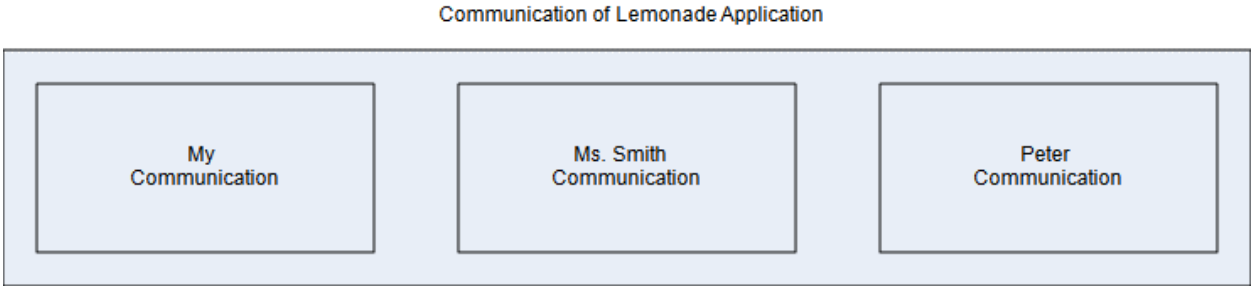
From the model above, we can see that to enable the making of the lemonade, I communicate with Ms. Smith about providing lemon for the lemonade only. To enable the making of the lemonade, I communicate with Peter about providing sugar for the lemonade only. From that, we can see that my communication, Ms. Smith's communication, and Peter's communication that drive the making of the lemonade. The combined communication is viewed as the natural communication for that lemonade. In this case, no extra communication nor unwanted communication is added to that communication in order to render the application complex.

External Communication Related to Complexity

Our application is a function of communication. If we have 5 people in our application, then it is driven by the communication of those 5 people. In this case, an outside person who is not in our application cannot communicate to drive that application. In other words, our application is naturally driven by 5 people who are part of that application. The communication of another person who is outside that application cannot communicate to drive that application. That application is naturally driven by people who are in that application, not by people outside that application.

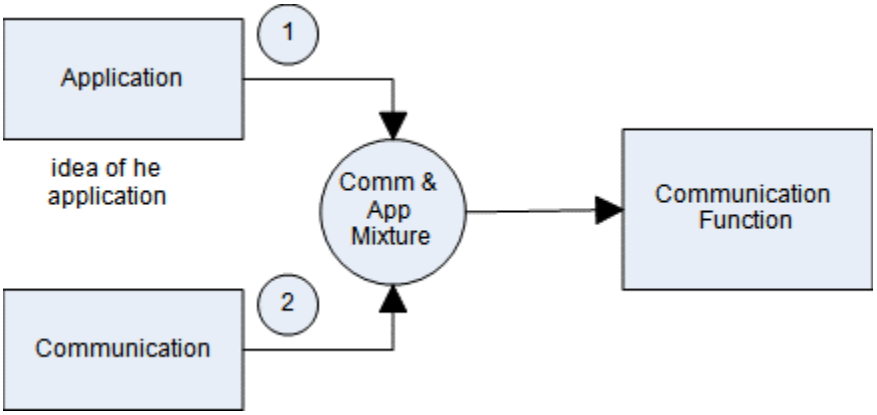
As we have learned previously, any extra communication added to that application by disregarding its natural form, enables the application to become more complex. In this case,

adding outside communication to that application enables the application to become more complex. Adding communication of another person who is outside that application to that application, simply adds extra or unwanted communication or ideas for that application which enables it to be more complex. For our lemonade application as shown by the model below, we can see that the overall communication for that application includes my communication, Ms. Smith's communication, and Peter's communication without any outside communication.



Since unwanted communication and unwanted ideas enable our application to become more complex, removing external ideas and communication from our communication enables us to remove unwanted ideas and communication which render our application to become less complex.

The increase of ideas in our application simply add complexity to our application. What do we mean by that? We mean ideas that are not naturally in our application. The way to look at it, naturally our application is defined by the model below.

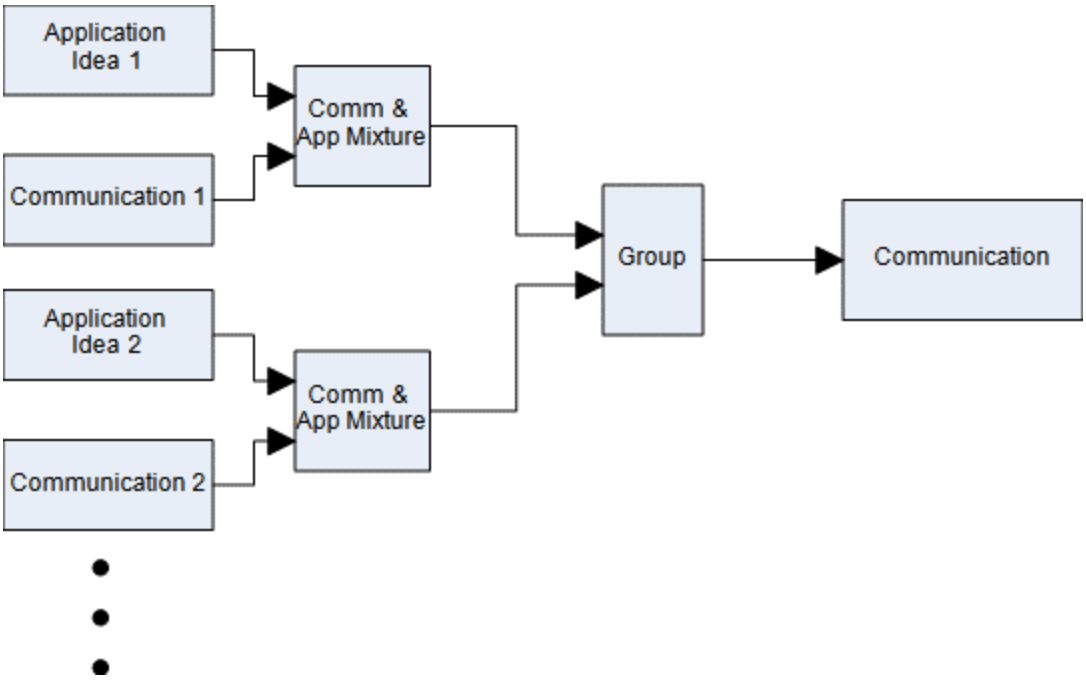


From the model below, at node 1, we can say that the idea of the application is natural for that application. At node 2, the communication is natural to that application. There are considered to be two natural entities that enable the function of the application to execute.

Now, let's identify another model, where the idea of the application increases by two more than its natural idea. In this case, naturally the application is executed by a single idea which is

 Copyright 2023 *The Speak Logic Project*

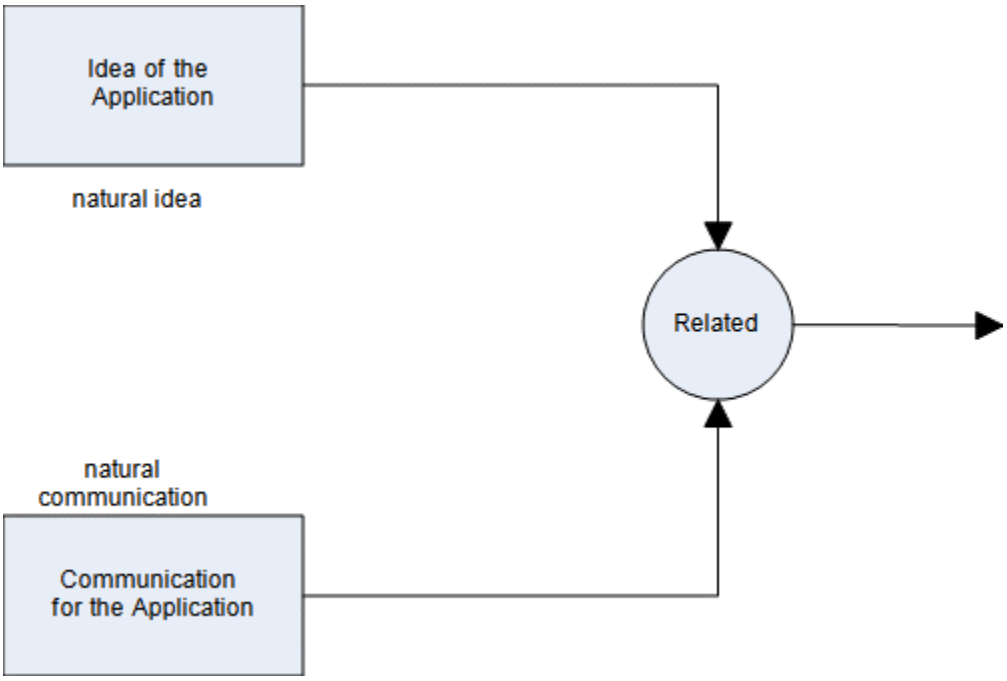
natural to it. Now, we increase the idea of the application to more ideas. We can represent that by the model below. We use the continuity to show more extras.



From the model above, we can see the increase of number of ideas in our application simply makes the application more complex. We can look at both models. For the first, we have the application is executed with its natural idea. For the second application, more ideas are added to the application. By simply looking at both models, we can see that the second model is more complex than the first one.

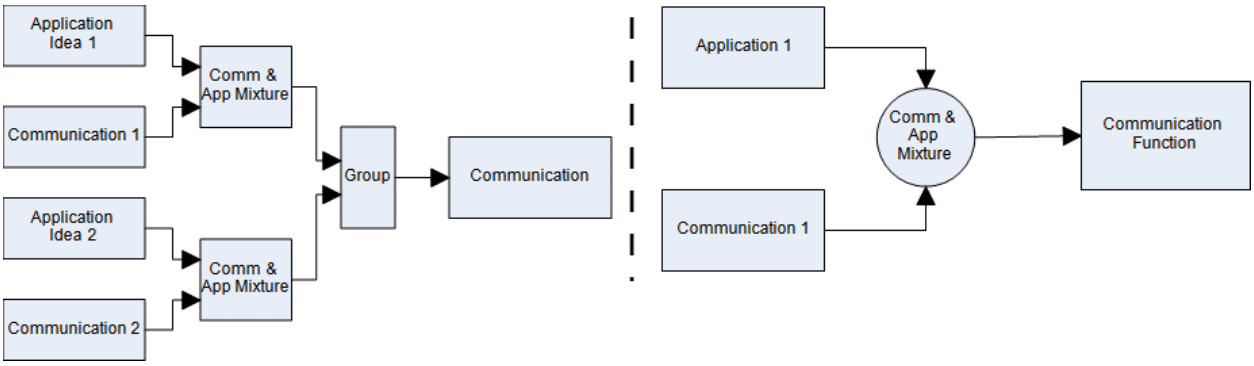
Now, in terms of complexity, we can determine whether the ideas are related or not. To better understand that, let's look at the communication for instance. If our application is driven by a single sentence, that sentence is natural for our application and we can see that it is a correct sentence because it enables the application as well as solves the underlying problem. The idea of that sentence related to that communication is also natural to that application. In this case, we can see that there is a relationship between the idea of the application and the communication for that application. We can represent that relationship in the form below.

 Copyright 2023 *The Speak Logic Project*



Now in terms of complexity added to our application by increasing number of ideas in our application, we can look at it two ways. First, we increase number of communications in our application; we also increase number of ideas in our application. If the communication contains errors that affect our application, then the idea of the application is bad. In this case, there is no relationship between the idea of the application and the communication for that application. This process simply increases complexity in our application. As shown from the model above, if the communication is correct, then it naturally drives the application and both of them are related. If the communication is incorrect, the idea is incorrect and both of them are not related to the natural communication which increases complexity in our application.

Another way to look at complexity in our application is to determine whether the extra communication is necessary to execute our application. To better understand this process, let's represent the two models we have identified above side by side.



 Copyright 2023 *The Speak Logic Project*

From the model above, we have Application 1 to the left as well as Application 1 to the right. From the one to the left, we have Idea 1 and Idea 2 where both of them are related. We also have Communication 1 and Communication 2 that are related to each other as well as the ideas of the application. Both Communication 1 and Communication 2 are error free and Idea 1 and Idea 2 are good. To the left, Communication 1 is error free and Application 1, which is the idea of the application, is related to that communication and it is good. Both models can be executed to solve the underlying problem. By looking at both models, we can see that the one to the left is more complex that the one to the right. In order to reduce complexity, it is better for us to choose the one to the right.

The way to look at it, even when the communication is correct, in order for us to reduce complexity it is always better to use the minimal communication. In other words, even with correct communication, reducing number of communications in our application further reduces complexity in our application. To better understand that, let's take it like this. If Communication 1 can drive our application to solve the underlying problem, where additional communication such as Communication 2, Communication 3, and so forth can also drive our application to solve the same problem, it is preferable to use Communication 1 only to drive our application in order to reduce complexity in our application.

The same applies, if the idea of the application is good. In order to reduce complexity in our application, it is always better to use the minimal idea. In other words, even with multiple good ideas, reduce number of ideas in our application further reduce complexity in our application. For example, if Idea 1 can be used with Function 1 to solve the underlying problem, additional ideas such as Idea 2, Idea 3, and so forth can also be used with Function 1 to solve the same problem. It is preferable to use Idea 1 only in order to further reduce complexity in our application.

For the lemonade application, the idea of the lemonade application is similarly represented by the communication model previously identified that represent communication of everybody in the application. In this case, the making of the lemonade is my idea, where providing lemon is Ms. Smith's idea, and providing sugar is Peter's idea. We can also say the lemonade is my idea, where the lemon is Ms. Smith's idea and the sugar is Peter's idea. What is important here is that in terms of communication, my communication to Ms. Smith is viewed as a single communication that enables the making of the lemonade. Ms. Smith's communication is viewed as a single communication that enables me to get the lemon to make the lemonade. Peter's communication is viewed as a single communication that enables me to get the sugar to make the lemonade. In this case, no additional communication is needed in order to reduce complexity of the application.

 Copyright 2023 *The Speak Logic Project*

Also, the idea of making the lemonade is a single idea without any extra ideas added. The idea of getting the lemon from Ms. Smith is viewed as a single idea without any additional ideas, so too is the idea of getting the sugar from Peter. In this case, reducing number of ideas in the lemonade application further reduces complexity in the application.

Application Performance Related to Complexity

Why it is important to reduce complexity in our application? Complexity in our application enables us to work hard where we result with less or nothing. We already know that the goal of our application is to execute a function to solve an identified problem. When we start our application, we need to work to make sure it is executed to solve the underlying problem. By introducing complexity in our application, we simply work harder for less or for nothing.

The questions we can ask are: If we can solve the underlying problem with minimal work, why work harder? If we can solve the underlying problem with minimal communication, why increase number of communications? If we can solve the underlying problem with minimal idea, why increase number of ideas?

The process of reducing complexity enables us to reduce number of communications in our application as well as number of ideas in our application. In this case, if the communication is not needed to solve the underlying problem, we simply discard it or not introduce it. As well, if the idea is not needed help us solve the underlying problem, we simply discard it or drop it from our application.

Since errors in communication enable our application to execute with errors, in this case we need to analyze our communication to reduce complexity in our application. In this case, when we analyze our communication and remove errors in our communication, we simply prevent errors from being introduced in our application. It enables us to reduce complexity in our application. By letting errors get to our application, our application will no longer solve the underlying problem. In this case, we simply work for nothing. By not removing errors in our communication, we simply work harder for nothing in return. Using the Error Correction Function to remove errors in our communication enables us to remove complexity in our application.

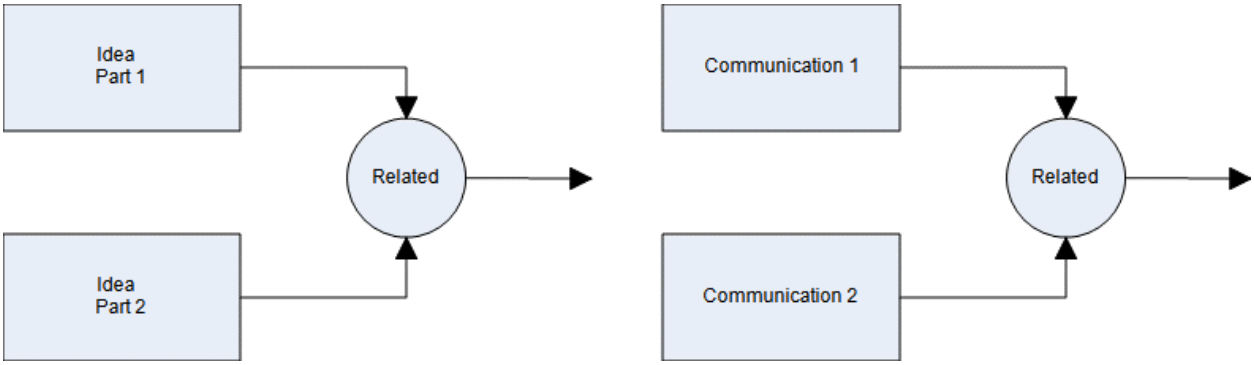
Our application exists to solve an underlying problem. A good idea for the application is the natural idea for that application to solve that problem. Any bad idea does not enable us to solve the underlying problem. In this case, a bad idea of our application enable us to work for nothing. This is the reason we need to discard bad ideas in our application. Bad ideas enable us to work for nothing rather than solving the underlying problem. By removing bad ideas in our

application or discarding them, we work to solve the underlying problem rather than working harder for nothing.

We can see the performance of our application in terms of reducing complexity in our application. In this case, the performance of our application is determined by the following: remove unwanted communication which enables our application to become complex. Discard unwanted ideas which enables our application to become complex. Correct errors in our communication with the usage of the Error Correction Function which enable us to work for nothing or work harder without solving the problem. Discard any bad ideas from our application which enable us to work harder without solving the underlying problem. Those things affect the performance of our application. When we follow the preceding steps, we simply increase the performance of our application.

Communication Relationship and Application Idea

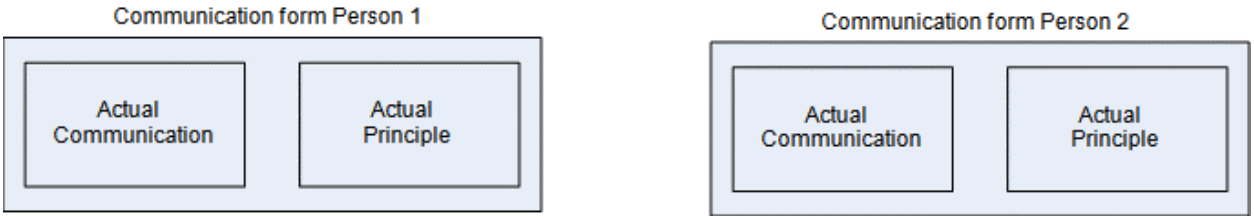
There is a relationship between the idea of our application and the communication of our application as we have shown previously. By understanding that, we can see if the idea of our application includes multiple parts, then the parts are related to each other in the form below. Also, if our application includes communication from multiple people, then the communication of the people is related in the form below.



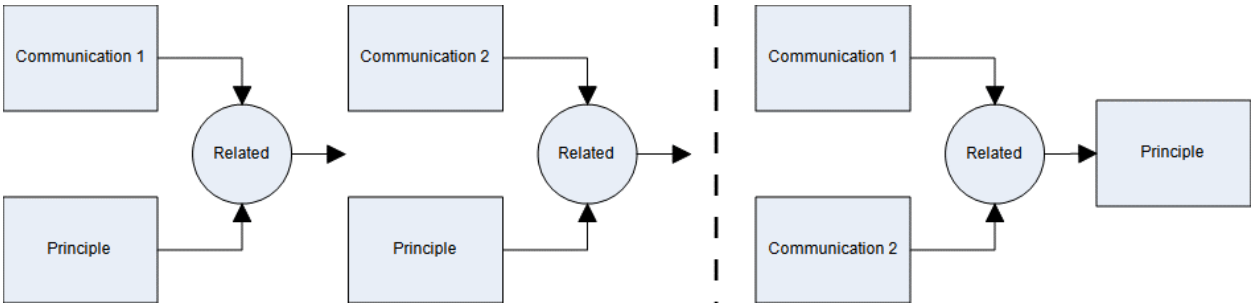
From the model above, both Part 1 and Part 2 of the application are part of a single idea. We can say that Part 1 and Part 2 are common for the application. Also, Communication 1 and Communication 2 are communication from two people who are in the application with both communication is common to each other. In terms of reducing complexity, we can see that common communication of the people in the application enables the reducing of complexity as well as common ideas from those people. In this case, in terms of reducing complexity in our application, we have to look at whether or not the communication of the people in the application are common to each other as well as the idea from each person which are parts of the idea of the application is common in the application.

Copyright 2023 *The Speak Logic Project*

In terms of reducing complexity in the application, we can determine whether the principle of communication is included in each communication. If the principle of communication is included in each communication, then there must be a similarity between the communication of the people in the application. In this case, the inclusion of the principle in the communication enables the communication of the people in the application to be common to each other. To better understand that, we can look at the model below.



From the model above to the left is the communication from Person 1 of the application while to the right is the communication from Person 2 of the application. As we can see, both sets of communication include the principle of communication. In this case, the inclusion of the principle in both sets of communication enable the communication to be common to each other. That relationship can be presented in the form below.



From the model above, we can see that the fact that Communication 1 from Person 1 is related with the principle and Communication 2 from Person 2 of the application is related to the principle, then both communications are related to each other by the principle. To the right, it shows the result of that relationship. In terms of reducing complexity, we can try to identify the principle in the communication of people of the application. In our analysis, we can determine whether or not the communication of the people is common to each other. If the principle is included in the communication of the people in the application, we can conclude that the communication is common, which enables us to reduce complexity in the application.

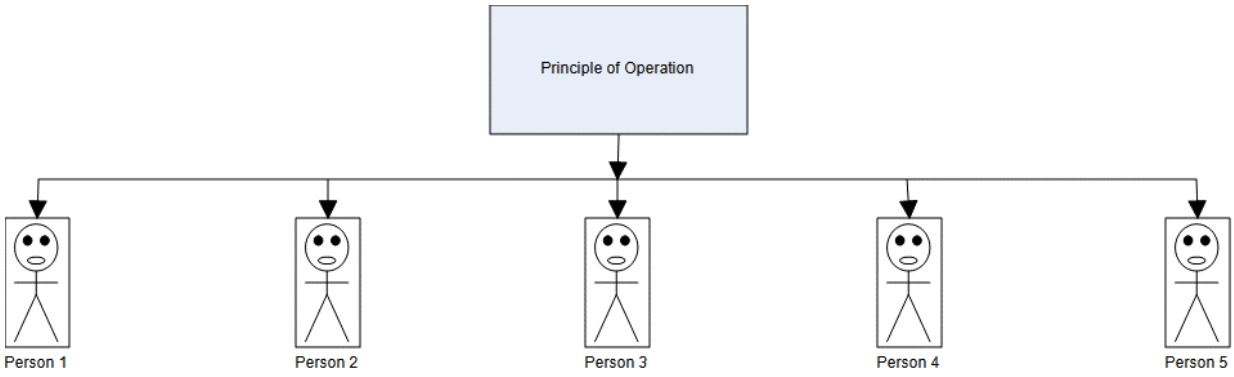
Operating Principle with Reducing Complexity

From the previous section, we have learned that common ideas and common communication in our application enable us to reduce complexity in our application. We have also learned that the

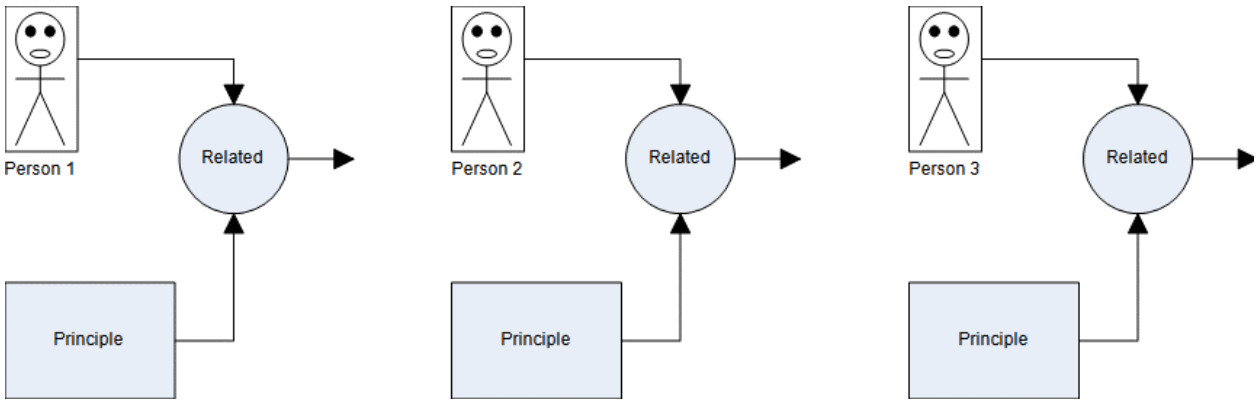
 Copyright 2023 *The Speak Logic Project*

common communication in our application is enabled by the principle. In other words, the inclusion of the principle in the communication enables the communication of people in the application to be common, which enables us to reduce complexity in the application.

Now, in terms of common entity between elements of the application, let's take a look at the operating principle of the application. For example, in terms of operating principle of the application, let's look at the principle of communication as one of the operating principles that enables us to execute our application. The way to look at it, our application is naturally driven by our communication. The communication that drives the application is common between the people in the application. The reason for that is because the communication of each person in the application includes the principle. In this case, we can identify the principle as a degree of operation of our application, or a high degree of operation of our application, or the highest degree of operation of our application. In this case, we can identify the principle in relationship with the people in the application in the form below.



From the model above, we can see that the principle of operation of the application is what the people in the application depend on to execute the function of the application. The fact that each person in the application depends on the same principle, we can see that the principle is common among them which enables the reduction of complexity in the application. To better understand that relationship let's represent it in the form below.

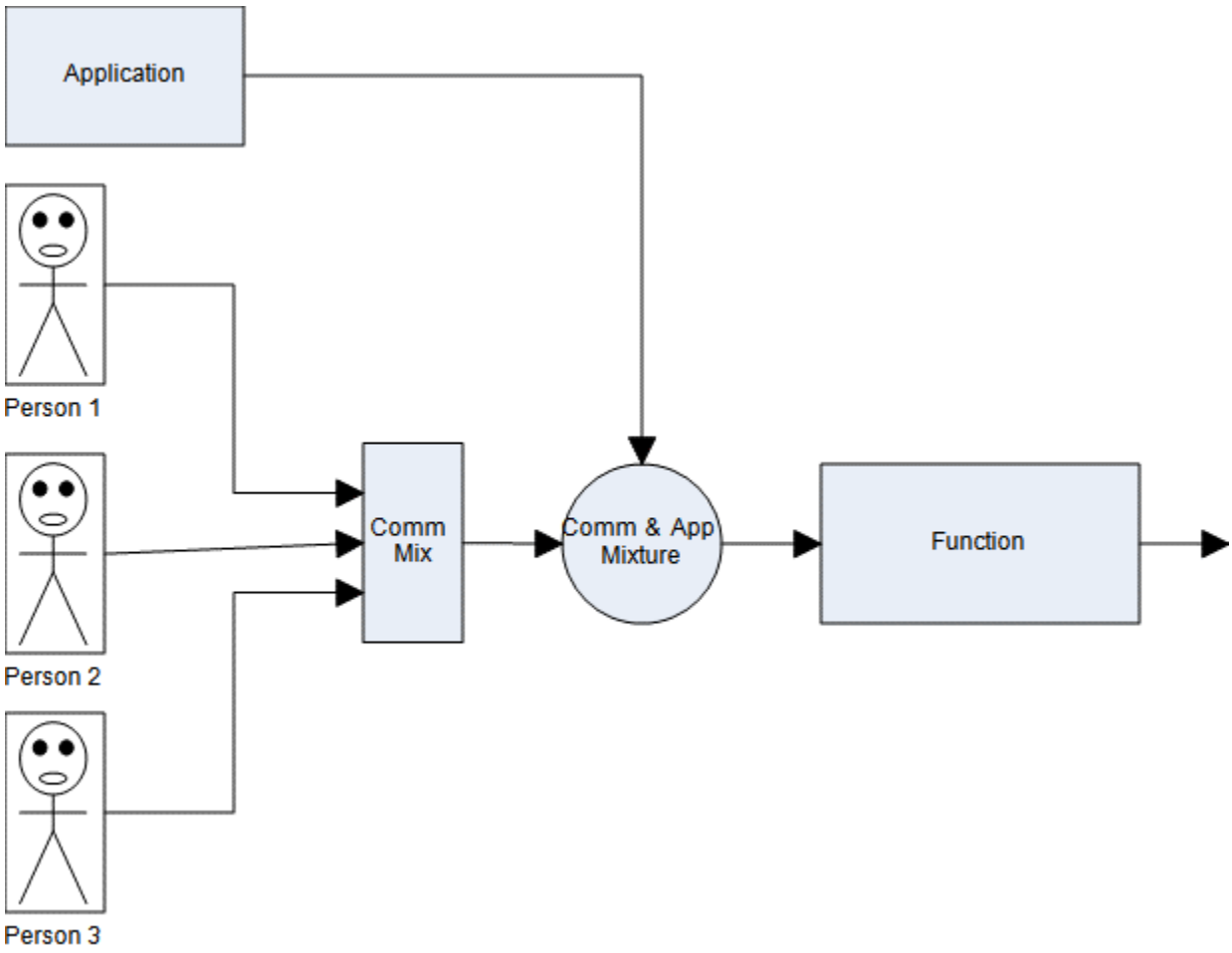


Copyright 2023 *The Speak Logic Project*

From the model above, we show only three people in the application for limited space. The model shows the principle is related to each person in the application. The common relationship among all the people in the application enables the reduction of complexity in the application. What is important here is that in our application, if we want to reduce complexity, we have to make sure the principle of operation is common among the people in the application.

Functions Discontinuation

Producing unneeded/unwanted or negative functions or force functions, to execute even when they are not needed don't solve any problem, also increases complexity. To better understand that, let's assume the function below is managed by Manager 1 or Person 1 in the form of:



If Person 1 or/and others have determined that the function does not solve any problem or are no longer needed, then the continuation of the execution can be stopped in order to reduce complexity. The way to look at it, if a function is negative or does not solve any problem, we simply stop the execution of that function. Once we continue to execute that function, we simply

 Copyright 2023 *The Speak Logic Project*

increase complexity in our application. From the model above, both the application and the communication of people in the application are viewed as negative.

Copyright 2023 *The Speak Logic Project*

Understanding Resource Management in the Communication Domain

Introduction

From the previous chapter, we have learned how complexity affects our application. We have also learned that since our application is naturally driven by our communication, our communication can increase complexity in our application. In terms of communication, in order to execute our application, we may need to use resources or elements to help us execute the function of that application. In the communication domain, we talk about resources that we can use to execute our application.

In the communication domain, we communicate about resources that we use to execute our application. Since our application is naturally driven by its communication, in this case we can say that the resources that we use to execute our application are naturally needed by our application. In this case, in order to execute the function of the application, those resources need to be used naturally in order for that to happen. Since those resources are the only ones needed by the application, we only need to communicate about those resources to drive the application. Any extra communication about extra resources that are not needed by the application, enables the increase complexity in the application.

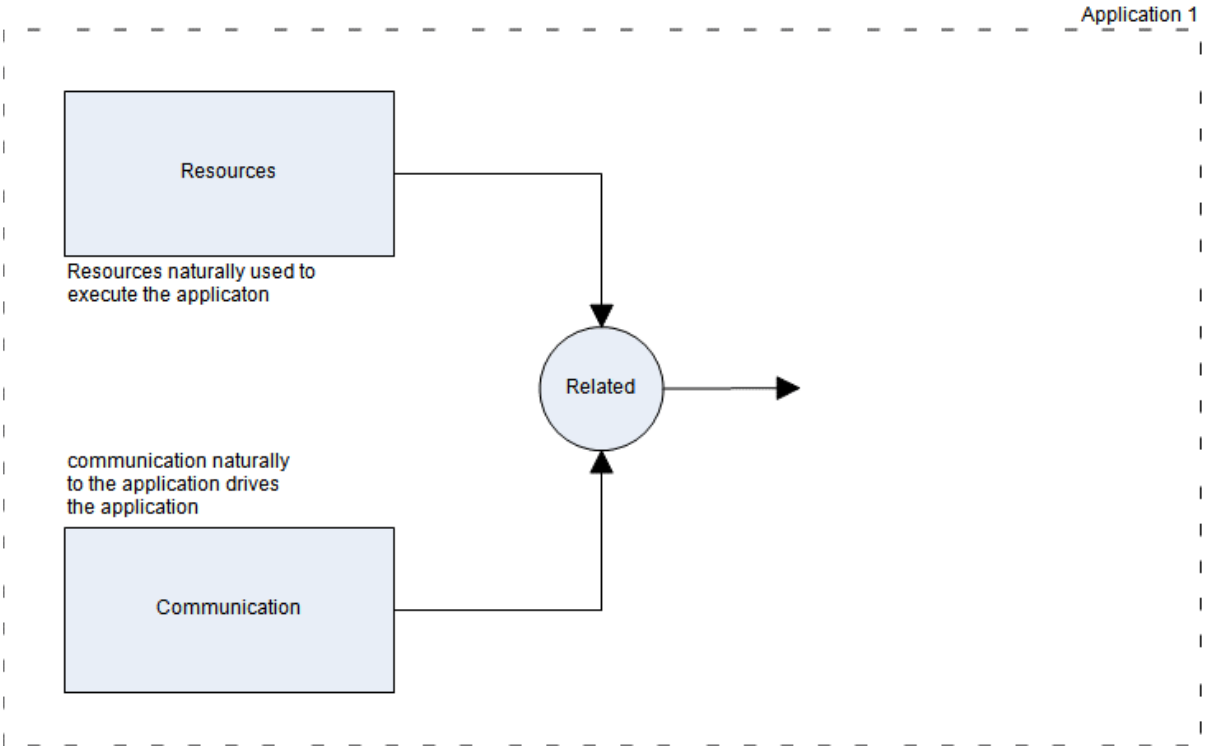
In this chapter, we are going to look at complexity of our application in relationship to resources usage in our application. In this case, we are going to look at how complexity affects our application in relationship to resources usage in our application and how we can reduce complexity in our application accordingly.

Copyright 2023 *The Speak Logic Project*

Complexity and Resource Management

In the communication domain, naturally our application is driven by our communication. In the communication domain, the function of our application is a function of communication where it is driven by its natural communication. The natural communication is defined as the only communication that is capable to execute the function of the application to solve the identified problem. We can say that communication is unique to that application without any extra added.

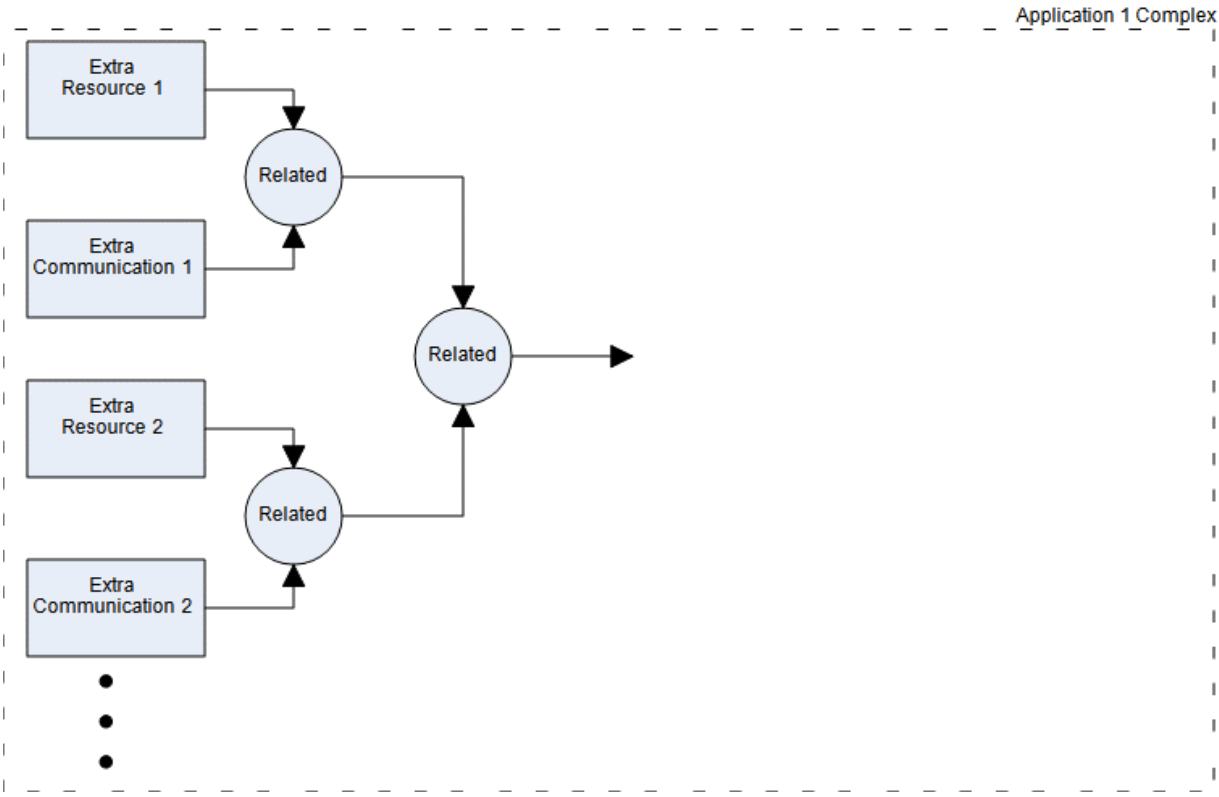
In the communication domain, in order to execute the application, it may require resources in order to do so. In this case, we communicate about resources that we use to execute the application. Here, resources are defined as entities that we used to execute the application, where the application is naturally executed by using the resources. We can say that those resources are needed by the application. In relationship with the communication, we can say that the application is naturally driven by its communication, where it is naturally executed by only resources that it needs. In this case, we have the following model.



As we can see from the model above, there is a relationship between the communication that is naturally driven by our application and the resources our application naturally needs to execute. In this case, any extra resources that are not needed by the application to execute simply increases complexity in the application. To better understand that, let's take it like this. For the application above, let's assume a single communication which is natural to that application and a single resource. Both execute together to solve the underlying problem.

Copyright 2023 *The Speak Logic Project*

Now, if we increase the number of resources, we also increase the number of communication. In this case, we can show the model in the form below.

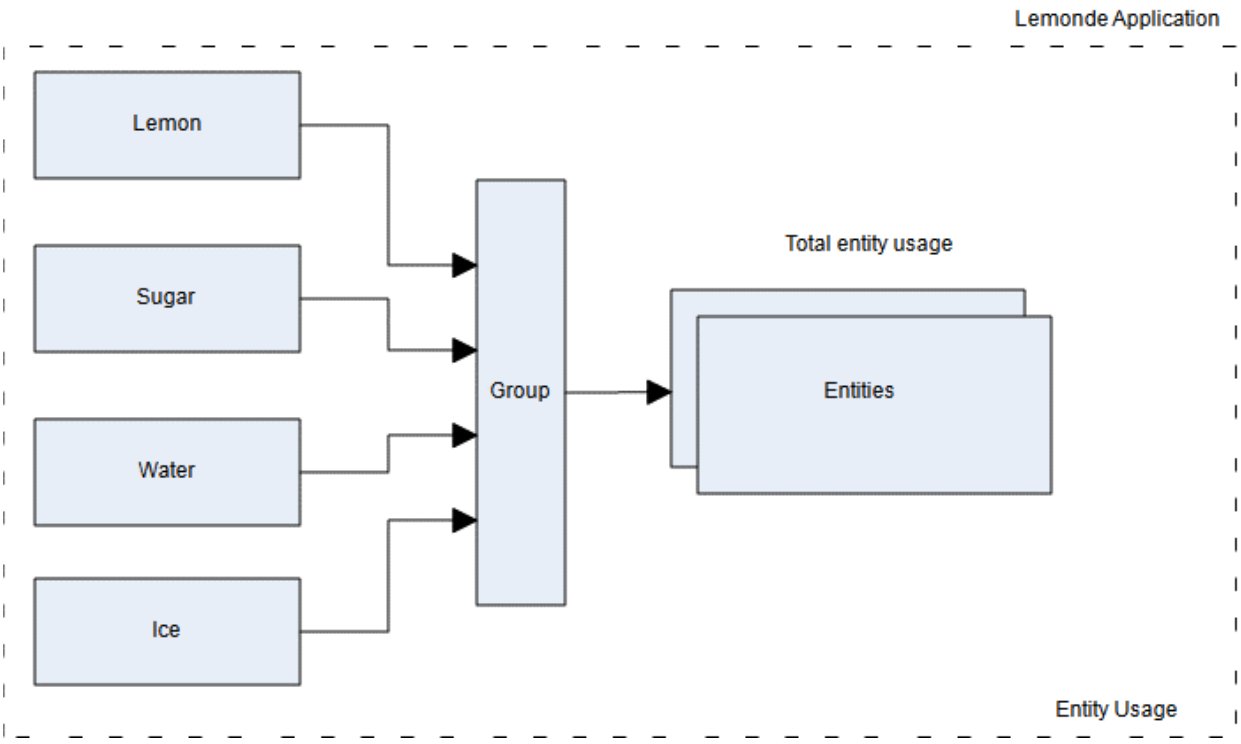


From the model above, we show two versions of Application 1. As we can see, by increasing the quantity of extra resources or entities to execute the application, we simply increase complexity in the application. As we can see by looking at both models, we can see that the second model is more complex than the first one. We use continuity in the second one to identify more that are not added because of space. What is important here is that in the communication domain, increased quantity of resources or entities that are not needed by the application to execute the application simply increases complexity in the application.

To better understand complexity or to better understand complexity in relationship to us, it is always good to define it as relationship. For example, an entity with fewer relationships is less complex than an entity with more relationships. From above, we can see that the first model of Application 1 is less complex than the second model, because it has fewer relationships. It is easier for us to understand an entity with fewer relationships while it takes more for us to understand an entity with more relationships. From that, we can see the increase of complexity in our application takes more effort from us. It enables us to work harder with less result as well as wasting resources, since the extra resources are not needed to solve the underlying problem.

 Copyright 2023 *The Speak Logic Project*

The reduction of unwanted or unneeded resources enables us to reduce complexity in the application. In other words, any communication or extra communication about resources that are not needed or wanted to execute the application, simply develops complexity in the application. We can also say that any communication or extra communication that is not needed by the application, simply increases complexity in the application. In this case, we reduce complexity in the application simply by not communicating about extra resources that is needed by the application to execute or solve the underlying problem. For example, for the Lemonade Application, the application is executed to solve the underlying problem by using the following:

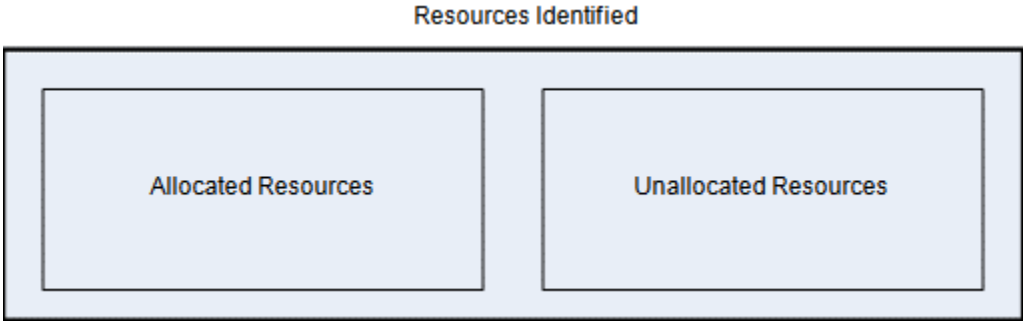


The model above shows the entity usage for the Lemonade Application. Since the application is naturally driven by its communication which is about entities that are needed by the application, the communication about the entities include: my communication, Ms. Smith's communication, and Peter's communication. From the entity usage model above, those entities are considered to be the only ones needed by the application. We can say that those entities are naturally needed by the application. In this case, the combined communication includes mine, Ms. Smith's, and Peter's only to enable the application without any additional or extra. Any extra entity that is not part of the application is excluded. Any communication that is not part of the application is excluded. Thus, it enables us to reduce complexity in the application.

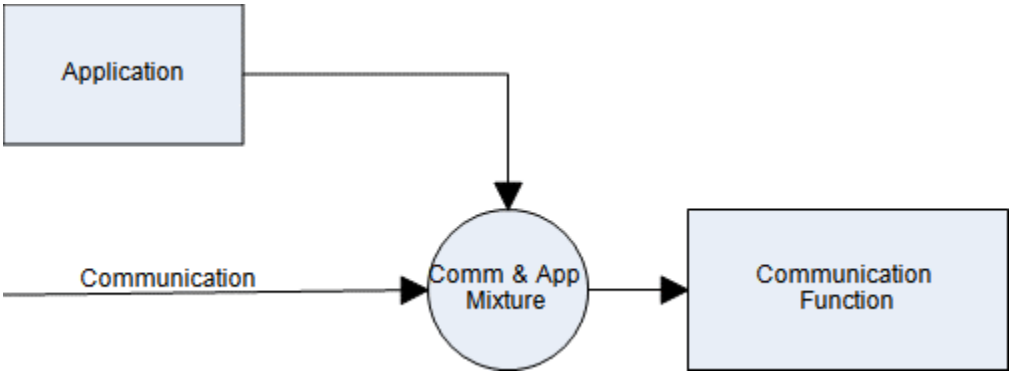
Copyright 2023 *The Speak Logic Project*

Visual Aspect of Communication and Resource Management

We identify or visually identify two sets of resources: one is allocated and the other one is not allocated as shown below.



From the model above, two sets of resources are identified. Each identified resource is viewed as an entity, where one is allocated and the other one is not allocated. In this case, the allocated resource is the resource we use in our application. The allocated resources are natural to our application and we can communicate about it. The communication about the unallocated resource does not enable our application. From that, we can see in our application if we communicate about the unallocated resource, we simply commit errors in communication which enables us to mismanage our resources. To better understand that, let's take a look at the model below.



From the model above, let's assume that in that project we have two people: Person 1 and Person 2, where the input communication above is viewed as communication of both Person 1 and Person 2 in the application. What is important here? Since that application is only enabled by the identified allocated resource and that communication is natural to that application, only that communication enables that application. The communication about the unallocated resources enables the application to execute with error, since that communication is not natural to the application. In other words, the allocated resource is the one that is natural to the application, therefore communication about that resource is natural to that application. It enables the

 Copyright 2023 *The Speak Logic Project*

application to execute without error. The unallocated resource is not natural to the application and communication about that resources is not natural to the application as well. When we communicate about the unallocated resources to execute the application, that enables us to commit errors in communication which enable the application to execute with error. The communication about unallocated resources enables mismanagement of resources and increases complexity in the application.

Problem Identification and Complexity

Proper problem identification is very important in terms of reducing complexity in relationship with resource management. By misidentifying the problem, we simply use resources to try to solve a problem, where the problem is not actually solved by the resources that we use. In this case, we simply waste resources by increasing complexity.

To better understand that, let's take it like this. Let's assume Problem 1 exists and our application exists to solve that problem. In this case, we use resources to solve that problem where we talk about the resources that we used to solve it. If we misidentify that problem, we simply communicate about resources that we need to solve a misidentified problem. In this case, the resources that we use simply wasted since the problem has been misidentified. The actual problem remains unsolved which enables the increase of complexity. In order to reduce complexity in our application, we need to identify problem properly. By identifying problem properly, we don't waste resources in areas that are not considered as problems or where problems are not identified. This process enables us to manage resources in our application.

Steps of Reducing Complexity with Resource Management

Some steps that enable us to reduce complicity in our application in relationship with resource management include:

- Proper problem identification: by identifying the problem properly, we don't use or waste resources in areas that are not the problem. In this case, we only use resources to solve the underlying problem when we properly identify it.

- Communicate only about resources or entities that are needed to solve the underlying problem or execute the function of the application. We exclude communication of extra(s) that is/are not needed to execute the application. In other words, any extra communication that is not needed to solve the underlying problem is excluded from our communication.

 Copyright 2023 *The Speak Logic Project*

- Focus enables us to reduce complexity in our application while disturbance enables us to increase complexity in our application. If we focus only on resources that enable us to execute our application, that enables us to reduce disturbances in our application. If we communicate only about resources that enable us to execute our application, we reduce disturbances in our application which further reduces complexity in our application.

- Focus only on the idea of the application without any extra added which enables the reduction of complexity in the application.

- If a function does not solve any problem or it is negative, we simply stop that function rather than continue to execute that function. When we stop that function, we reduce complexity as well as manage our resources. For example, if in our application we have multiple people where the combined function is managed by a manager and that function is negative or does not solve any problem, if we continue to execute that function we simply increase complexity where stopping executing that function enables us to reduce complexity. When we stop executing that function, we simply manage our resources.

Copyright 2023 *The Speak Logic Project*

Understanding our Parent's Vocabulary

Introduction

From the error correction process, we have learned that the feedback provided to us by our parent enables us to correct our communication errors. By taking a quick look at the process, we can see that, in some instances, the correction process happens because we use words that we are not supposed to use. In this case, we received feedback from our parent to enable us to change those words and use proper words.

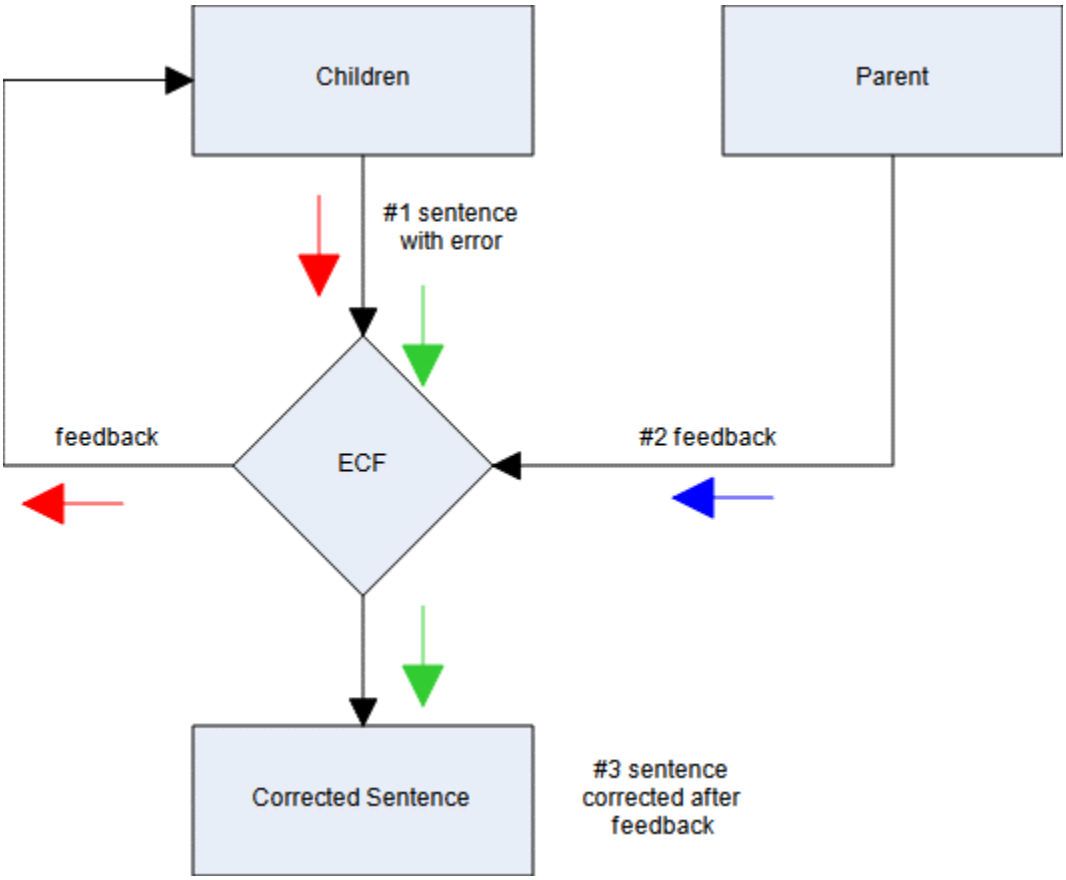
By understanding the paragraph above, we can ask these questions: Why does our parent think we are using unnecessary words or bad words? Why does our parent give us feedback when we use such words? From that, we can see that it looks like those words are not part of our parent's dictionary or vocabulary. By understanding that, we can ask this question: If those words were in our parent's dictionary or vocabulary, why does our parent think such words are not good? In this case, it looks like our parent does not use such words.

In this chapter, we are going to look at words that we use compared to words used by our parent. Since our parent does not use certain words, it looks like the size of our parent's vocabulary is much less than the size of ours . In this chapter, we are going to look at the size of our dictionary compare to the size of our parent's and provide many techniques that enable us to correct our own sentences rather than relying on our parent. In this case, this chapter enables us to be self-sufficient in terms of correcting our errors rather than depending on our parent for corrections.

Copyright 2023 *The Speak Logic Project*

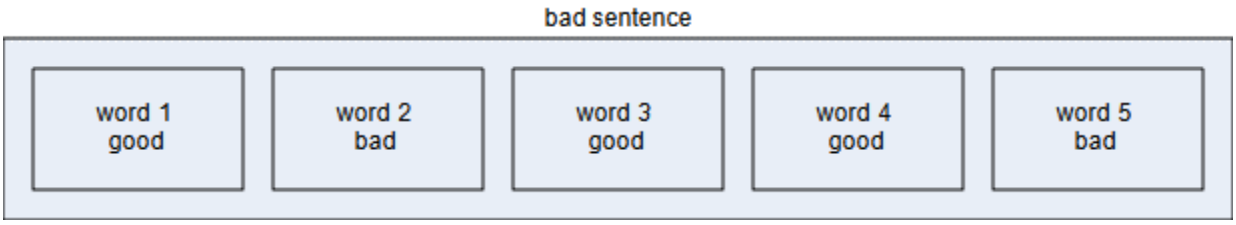
Understanding our Parent Dictionary

We can also say that **Understanding our Parent Vocabulary**. By taking a closer look at the feedback process related to our parent, we know that as we are talking our parent listens to us. During this process, if we repeat bad words, or unportable words, or words that point to wrong entities, or do not point to actual entities, we receive feedback from our parent to correct us to repeat proper words. By looking at the overall process, we can see that our parent's vocabulary or dictionary is different than ours. In other words, by looking at the process we can see that our dictionary contains words that are not in our parent's dictionary. To better understand the overall process, lets' take a look at the model below with the inclusion of the Error Correction Function (ECF).

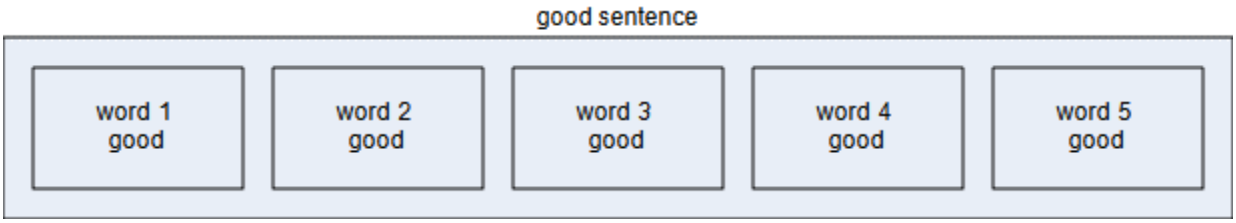


To better understand the model above, let's take it like this. At number 1, we repeat a sentence with errors where that sentence is viewed as a collection of words. In this case, that sentence is represented in this form:

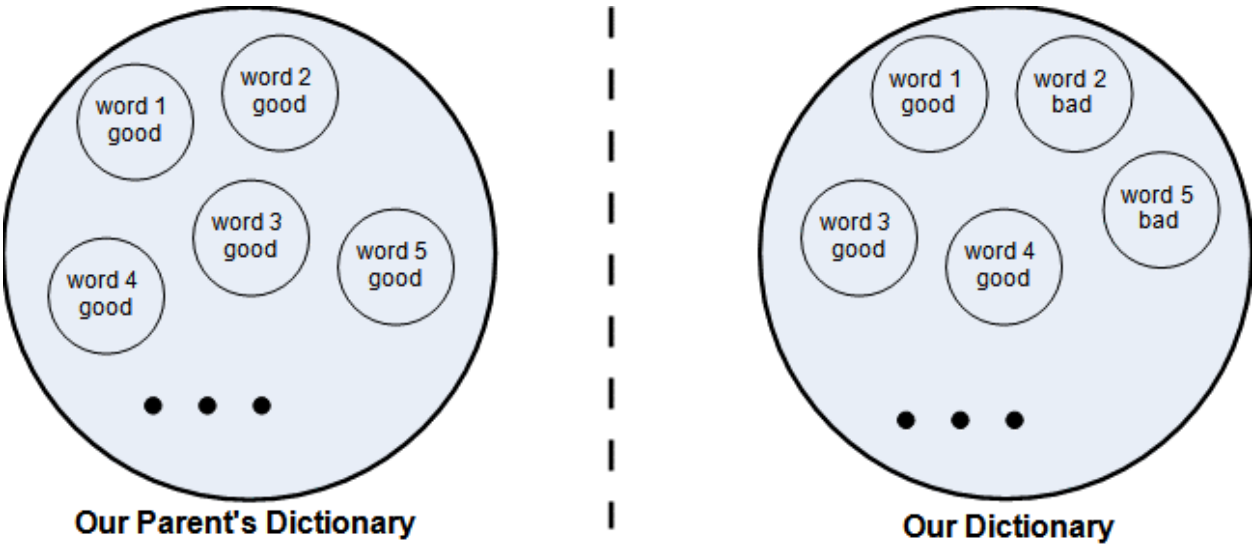
 Copyright 2023 *The Speak Logic Project*



After applying feedback from our parent, our corrected sentence looks like:



From above, we can see that our parent's dictionary has no negative words, while our dictionary does. From that, we can represent both our parent's dictionary and our dictionary below. From the model below, we use the triple periods to show continuation of more words.



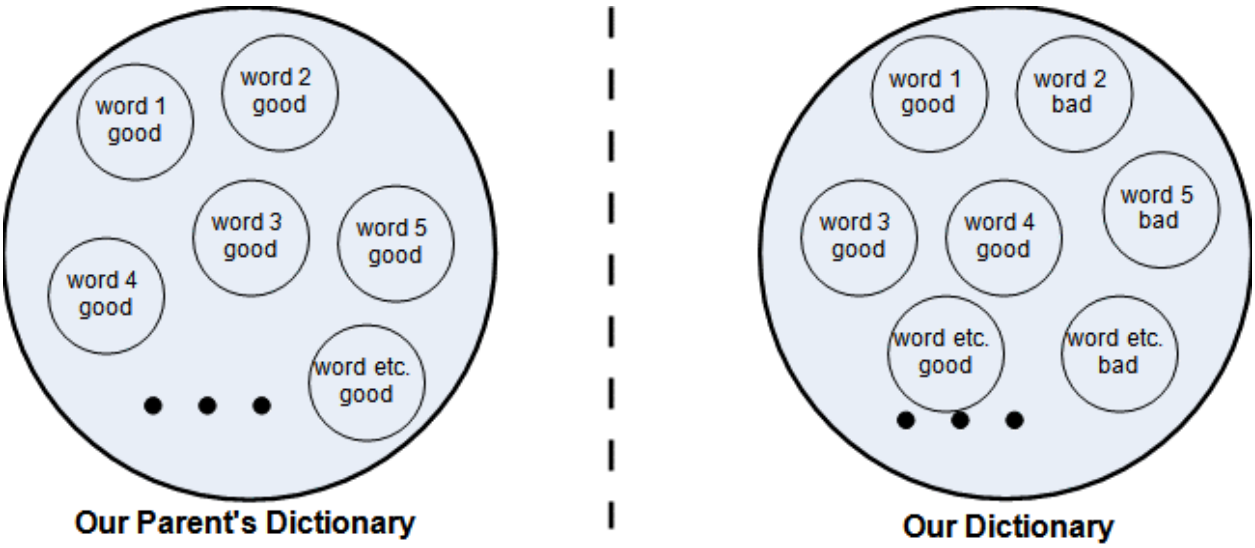
The words dictionary and vocabulary here are the same or interchangeable. The usage of the word dictionary here does not have to be a written dictionary. It is simply a collection of words that we use in our communication. That collection does not have to be a written collection.

Reduction of Words in our Dictionary

From the explanation above, we can observe in terms of size, our parent's dictionary is much more compact than our dictionary. We can also say that our parent's dictionary is small and

 Copyright 2023 *The Speak Logic Project*

portable, while our dictionary is much bigger and unportable. From that, we can represent both our dictionary and our parent's dictionary in this form.



From the model above, in both our parent's and our dictionaries, we use *word etc. good* for more words that are not shown in the model and in our dictionary we use *word etc. bad* for more bad words that are not shown in the model.

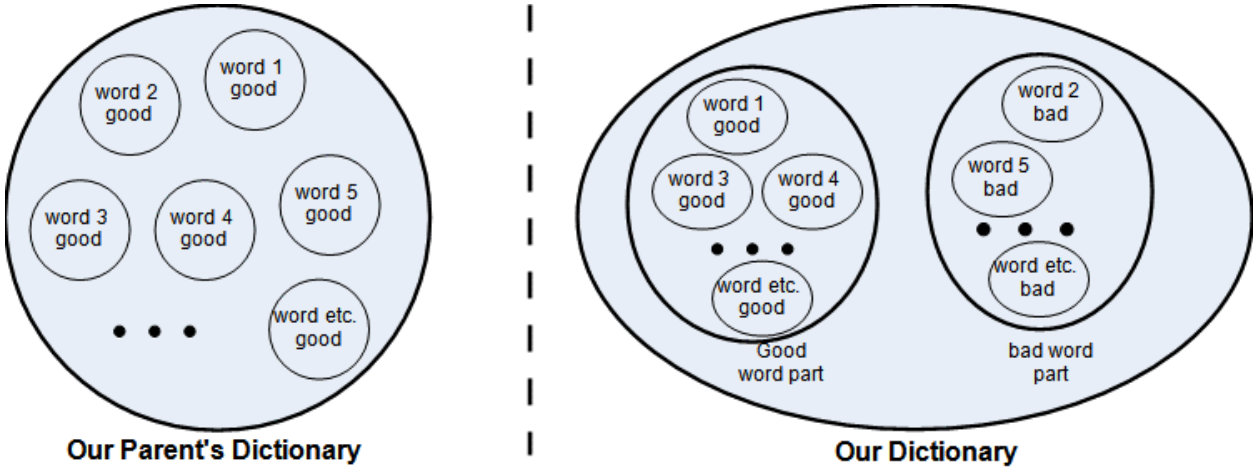
From above, since our parent's dictionary does not contain bad words, we can see that the size of our parent's dictionary is smaller than the size of our dictionary. In other words, since our dictionary contains both good words and bad words, the size our dictionary is much bigger than the size of our parent's dictionary.

Now, in order to make progress in our communication, we need to reduce the size of our dictionary to match our parent's dictionary. In other words, in order to be self-sufficient not to depend on our parent all the time to give us feedback to correct our communication, we need to reduce the size of our dictionary to be equal to our parent. By reducing the size of our dictionary to be the same as our parent, that enables us to remove all negative words that are in our dictionary as well as words that do not point to actual entities and words that point to wrong entities and so forth. Here we are going to provide multiple techniques that enable us to reduce words in our dictionary so our communication can be very portable.

Our goal is to reduce the size of our dictionary to match our parent's dictionary. As pointed out above, the size of our dictionary is much more than the size of our parent's dictionary. Here we are going to provide multiple ways to reduce the size of our dictionary to match our parent's dictionary so our dictionary can be very compact and portable. By reducing the size of our dictionary to make our dictionary portable, this also makes our communication very portable.

Copyright 2023 *The Speak Logic Project*

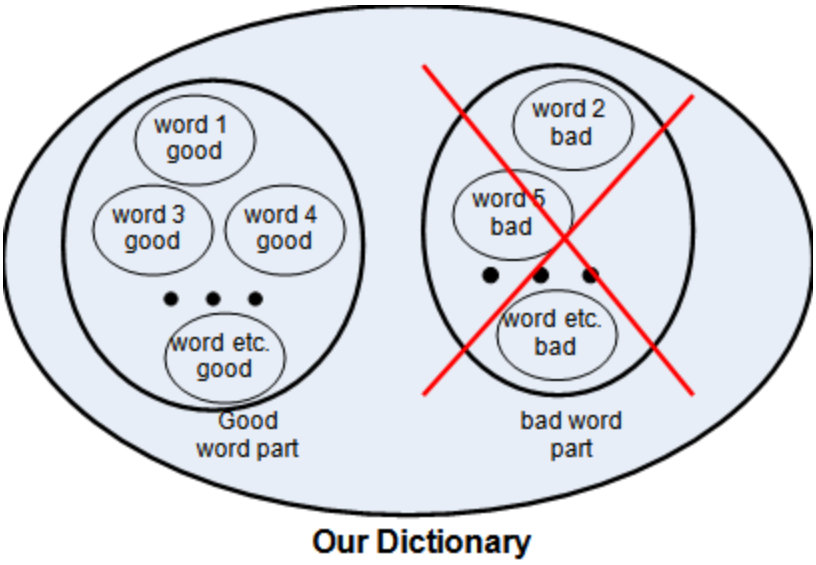
The overall process of reducing the size of our dictionary can be viewed in this form. Right now, our dictionary contains both bad words and good words, while our parent's dictionary contains good words only. By understanding that, we can see we have 3 separate entities, in terms of dictionary. We have our parent's dictionary as one entity, we have our dictionary as another entity. Since our dictionary contains both bad words and good words, it can be viewed as two separate entities or as two separate parts. One part is a good words part and another part is the bad words part. In this case, we can see that we have two separate entities overall: one entity is our parent's dictionary, another entity is the good words part of our dictionary, and the other entity is the bad words parts of our dictionary. We can represent them by the sets below.



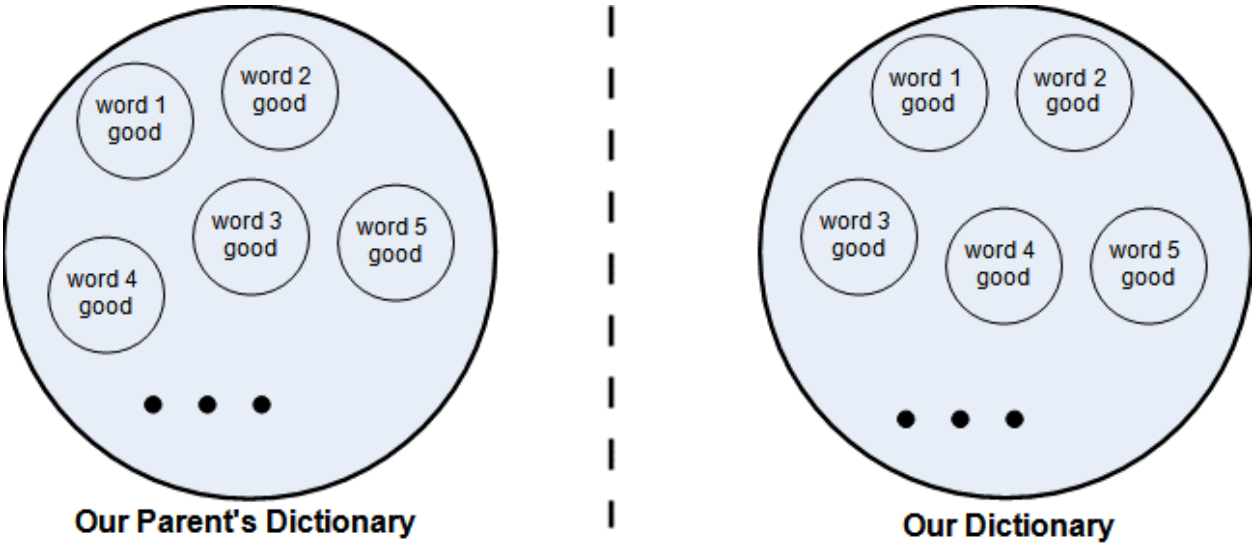
Since our dictionary contains both bad words and good words, we grouped all of the good words into one part and all of the bad words into another part. The results give us two separate entities: a good part and a bad part as shown to the right of the model above.

By understanding the overall explanation up to here, we can see that we have 3 dictionaries: our parent's dictionary shown to the left above, the good part of our dictionary which is viewed as a dictionary by itself and the bad part which is viewed as another dictionary. We can also say the bad part is a sub dictionary of our dictionary. In terms of number or limit, since the size of our dictionary is bigger and contains both positive and negative words, it has more words than our parent's dictionary or its limit goes higher. Our goal is to reduce the size of our dictionary. By reducing the size of our dictionary, we need to eliminate all bad words from our dictionary. For example, if we cross out all bad words in our dictionary or eliminate them as shown by the model below, then we have only good words left in our dictionary.

Copyright 2023 *The Speak Logic Project*



Crossing out bad words will eliminate all bad words from our dictionary, and we will have only the good words part of our dictionary left. In this case, we can see that our dictionary will look like our parent's dictionary. By eliminating bad words in our dictionary, we can say that our dictionary is produced from our parent's dictionary or is a product of our parent's dictionary. Since our dictionary is produced from our parent's dictionary, in this case we can say that it is equal to our parent's. By understanding that, we can represent our dictionary in this form.



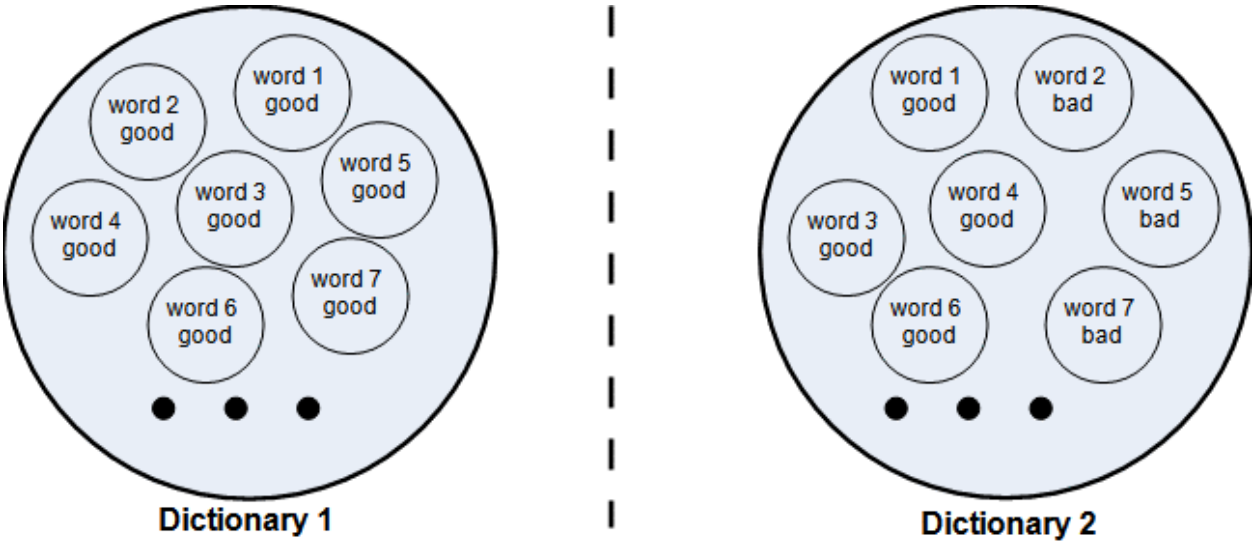
From above, by letting our dictionary equal to our parent's dictionary, all the bad words from our dictionary have been eliminated which makes our communication very portable. By letting our dictionary equal to our parent's dictionary, all words from our dictionary are portable and it becomes very compact and portable. Here, we are going to provide a lot of examples on how to

 Copyright 2023 *The Speak Logic Project*

accomplish that. We will provide examples and how to make our dictionary equal to our parent's dictionary so our communication can be portable. That will also make us self-sufficient.

Size of Dictionary Question

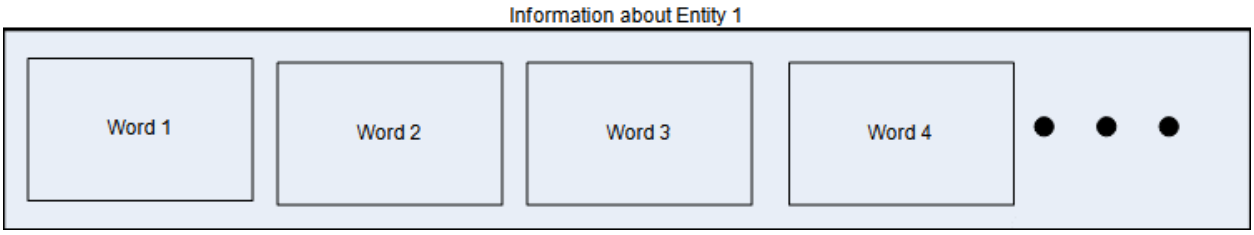
Here, let's ask and answer this question: Why are not too many words needed in a dictionary? As an example, let's assume that we have two dictionaries as shown below.



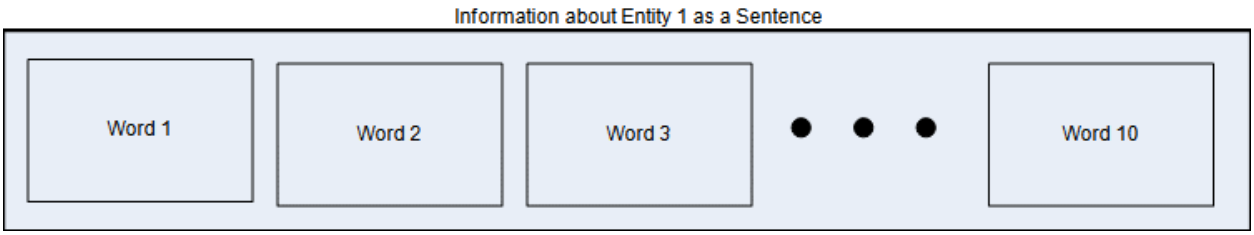
In this case, the size of Dictionary 2 is bigger than the size of Dictionary 1. We can also call Dictionary 2 a dictionary of bad words, since it contains both good words and bad words.

We already know that we communicate to each other by passing information or providing information to each other. In other words, our communication interface enables us to communicate to each other by providing information about entities that we identify. We know that the communication about an entity points to that entity. We know that the information about an entity depends on that entity. We know that the information an entity points to that entity. We also know that our communication exists after information. In other words, after we identify information about an entity, then we can communicate about that entity. In this case, the information about an entity exists first, where the communication about an entity exists second after that information. From what we have just said, we can see that if we identify *Entity 1*, then the information about entity 1 can be viewed as a collection of words as shown below.

 Copyright 2023 *The Speak Logic Project*



Here we represent the information about Entity 1 as a combination of words. If we want to, we can represent that information as a group of sentences. Now, let's represent that information as a sentence that is made of 10 words. In this case, we have:



From above, the information about *Entity 1* is viewed or represented as a sentence of 10 words. In this case we can see that all words in that sentence belongs to the dictionary. In other words, that sentence belongs to the dictionary because all words in that sentence belongs to the dictionary. For example, Word 1 belongs to the dictionary, so do Word 2, Word 3, and so forth.

Since information about Entity 1 depends on Entity 1 and all words that represent or describe that information belongs to the dictionary, then there is no need to have negative extra words in that dictionary or there is no need to have a separate negative dictionary. In this case, we can see that it is better not to have too many words in a dictionary, because it enables us not to add extra to information about entities. Another way to say it is that since information about an entity points to that entity and depends on it, we don't really need extra words to present or describe that information. By not having too many words in our dictionary, it helps us with the following:

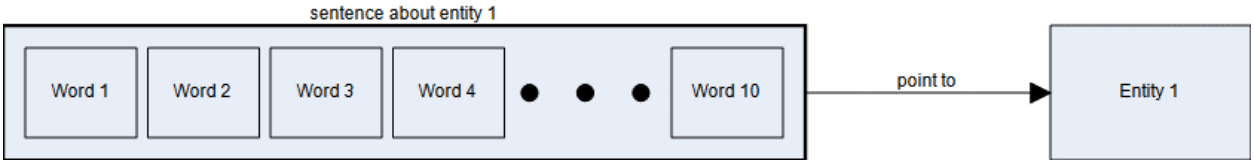
- It enables us to communicate relatively about entities without adding ourselves to it. In other words, it enables us to provide information about entities without adding ourselves to that information.
- It enables us to provide information without adding negativity to that information. In other words, we can provide information about entity where the information depends on that entity rather than us. In this case, it is not possible to add negativity to that information, since that information does not include any negativity.
- It enables us to provide information, without adding our feeling to that information since that information is not a part of our feeling. In other words, by not having too many words in our dictionary, that enables us to control our feeling without the inclusion of adding our feeling to information about entities.

 Copyright 2023 *The Speak Logic Project*

- It enables our communication to be portable. By not having too many words in our dictionary, that can make our communication more portable. In other words, by not having too many words in our vocabulary, that make our communication more portable.

Another way to look at it, there is no need to have too many words in our dictionary. Since we communicate relatively about entities that we identify, during our communication, if we don't identify an entity, we cannot talk or communicate about that entity. According to our parent, there are many words that we use that don't exist. In this case, those words don't point to actual entities or identify actual entities. By removing those words from our dictionary, that improves our communication much better. The way to look at it, if *Word 2 Bad*, *Word 3 Bad*, and *Word 5 Bad* exist in a dictionary and they don't point to actual entities or identify them, then there is no need to have *Word 2 Bad*, *Word 3 Bad*, and *Word 5 Bad* in that dictionary. In other words, if those words exist in our dictionary and they don't identify actual entities or point to actual entities, there is no need to have them in our dictionary. We can simply remove them. By removing them from our dictionary or our vocabulary, that reduces the size of our dictionary or vocabulary and makes our communication much better and more portable.

Another way to look at the example above is that communication about an entity points to that entity. In this case, communication about Entity 1 points to Entity 1. As a separate entity from Entity 1, we can define communication about Entity 1 as a single sentence like that is made of 10 words, where all of them point to that entity as shown below.



What is important here is that since all words in that sentence belong to the dictionary, there is no need to have extra words to communicate about *Entity 1*

Since our commination comes after the existence of *Entity* 1, all communication about Entity 1 points to Entity 1. In this case, all words that are used for communication about *Entity 1* belong to dictionary as well.

Another way to look at it is that if we have a big dictionary and our communication is not portable, our big dictionary serves us no purpose. If the size of our dictionary is big and our dictionary is not portable, that dictionary serves us no purpose. If we have a big dictionary and our parent still corrects our communication, that dictionary serves us no purpose. If the size of our dictionary is big and we still cannot communicate with our parent, that dictionary serves us no purpose. Therefore, it is good for us to reduce the size of our dictionary in order to make our

 Copyright 2023 *The Speak Logic Project*

communication portable. It makes sense for us to reduce the size of our dictionary, so our parent does not have to correct our communication. It makes sense for us to reduce the size of our dictionary so we can communicate with our parent.

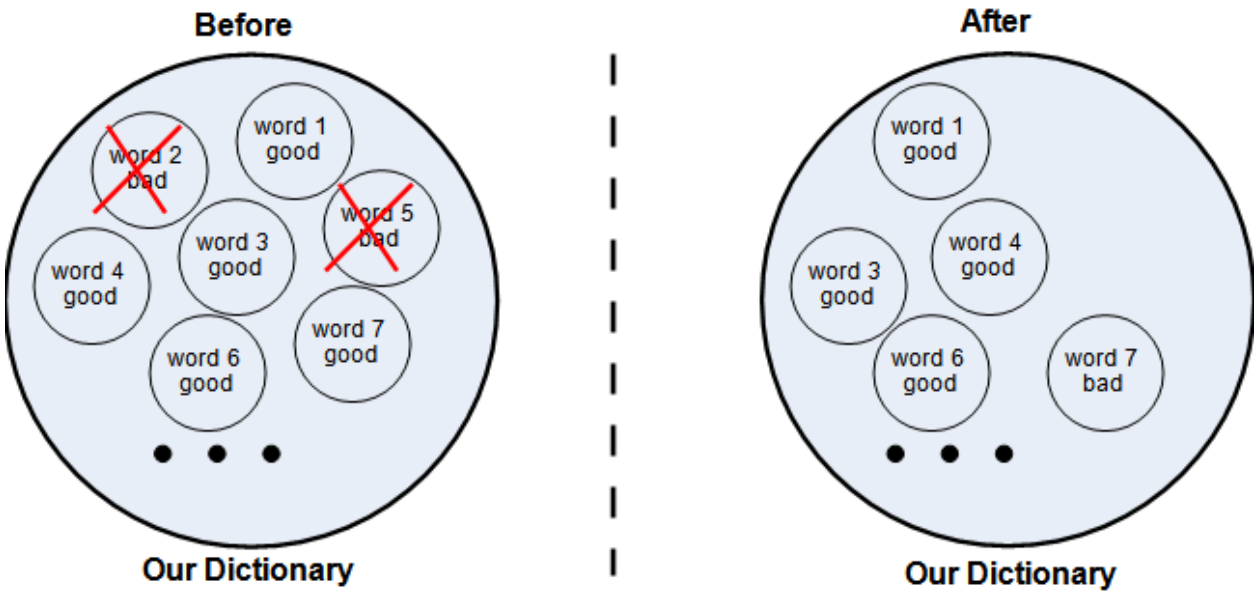
We can also ask this question: Why is the size of our dictionary getting so big? The way to look at it, as we keep thinking negatively, we start adding more words to our dictionary. For example, a negative idea enables a new negative word to be added to our vocabulary or dictionary. Based on that word, new words can also be added or derived. This process enables the expansion of our vocabulary. It enables our vocabulary to grow to a bigger size. This process also applies to entity misidentification.

Why have so many words in our dictionary if they don't help us solve any problem? By looking at the feedback model in relationship with our parent, we can see that we don't need too many words in our dictionary in order to solve problems. From the feedback process, we can see that the more words we have in our dictionary, the less likely we can solve problems with them. In the feedback model, we see that our dictionary has more words than our parent's dictionary. From the feedback model, we can see that the more words we have in our dictionary, the less likely we will solve problems with them. In this case, as we keep reducing the number of words in our dictionary, we solve more problems. We can see that there is a relationship between the size of our dictionary and problems. As number of words in our dictionary increases, we have more problems as the number of words decreases. We solve problems or have less problems. Therefore, it is important to reduce the size of our dictionary because it enables us to solve problems. Increasing the size of our dictionary is not important for us because it does not enable us to solve problems.

Example Number 1

Since our dictionary contains only good words and does not contain bad words, we can see that. *If a good dictionary exists, that dictionary cannot contain bad words*. In other words, if a dictionary contains good words only, that dictionary cannot contain bad words. In this case, if we identify bad words in that dictionary, we simply remove them since that dictionary cannot contain bad words. As an example, let's represent that in the model below.

 Copyright 2023 *The Speak Logic Project*



As shown by the model above, since the existence of our dictionary cannot contain bad words, we simply eliminated them from our dictionary. Doing so enables us to reduce the size of our dictionary. As shown above to the left, our dictionary contains bad words which increases the size of our dictionary. To the right, we can see that eliminating those bad words from our dictionary by crossing them out, we simply reduce the size of our dictionary.

By understanding the relationship of words in our communication, we can see that *if at least one good word exists in our dictionary, then there should not be any bad words in our dictionary*.

In this case, we can cross out words to denote words that do not exist in our dictionary. For example, if *word 2* is bad, we cross it out to show it does not exist in our dictionary. The table below shows an example of the elimination process.

Original	Process of Elimination
Word 1	Word 1 is valid, so we keep word 1 in our dictionary
~~word~~ 2	~~Word 2~~ is not valid, therefore , we cross it out so it is eliminated from our dictionary
Word 3	Existence of *Word 3* is valid, so we keep it in our dictionary
Word 4	Existence of *Word 4* is valid, so we keep it in our dictionary
~~word~~ 5	Existence of ~~Word 5~~ is not valid, so we remove it from our dictionary

 Copyright 2023 *The Speak Logic Project*

After the elimination process, we can see that the size of our dictionary is reduced.

What is important here is that existence of positive words in our parent's dictionary does not allow the existence of negative words. By understanding the feedback process, we can see that when we repeat bad words, our parent provides us feedback to enable the correction. This process tells us that our parent's dictionary contains good words only and does not contain any negative word. Since we want our dictionary to be equal to our parent's dictionary by containing good words only, the process of elimination enables us to analyze words and determine that they are bad and eliminate them from our dictionary so we can match our parent's dictionary. From above, we can simply re-index them.

Now that we understand our dictionary must contain only good words in order to be equal to our parent's dictionary, we can continue with the overall process of reducing the size of our dictionary further. From now on, all the other examples will be based on the first example.

Example Number 2

If a word can be combined with another word to form a new word, then that new word may not be needed. This is the same as saying: *If Word 1 can be combined with Word 2 to form Word 4, then Word 4 may not be needed*.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 1, Word 2, Word 3, Word 5, etc.

From the table above, by removing *Word 4* from our dictionary that further reduces the size of our dictionary.

Example number 3

If a word exists in our dictionary and it has been misused or caused confusion, then it can be replaced by another word. This is the same as, *if Word 1exists in our dictionary, where the usage of Word 1 has been causing confusion or Word 1has been misused, then Word 1 can be replaced by another word or combination.*

By applying this technique above, then *Word 1* can be replaced by *Word 2* or *Word 2* in combination with *Word 3*.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 2, Word 3, Word 4, etc.

 Copyright 2023 *The Speak Logic Project*

From the table above, by removing *Word 1* from our dictionary which is viewed as the confused word, that further reduces the size of our dictionary. In this case, we can see that if a word has been misinterpreted or misused, it is better to replace it with its actual meaning or combination of words or terms or expression.

Example Number 4

If a word or the definition of a word has already been defined or implemented in another word or a group of words, then the usage of that words can be dropped; especially if that word has been misused.

We can also put it in this form:

If Word 1is already defined in Word 2 or in a combination of Word 3 and Word 4, then Word 1 can be dropped from our dictionary and replaced by Word 2 or a combination of Word 3 and Word 4; especially if Word 1 has been misused.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 2, Word 3, Word 4, etc.

That will further reduce the size of our dictionary. The way to look at it, since *Word 1* is already included in *Word 2 or Word 3 and Word 4* combined, in this case *Word 1* can be dropped from our dictionary. By dropping *Word 1* from our dictionary, *Word 1* no longer exists in our dictionary. In this case, rather than using the actual word, we can simply use the actual definition or refer to the actual definition of the actual word or the combination of words instead.

Example Number 5

If the structure of a word is already included in another word or a combination of words, then that word can be dropped; especially if that word has been misused or misinterpreted or misunderstood.

We can also put it in this form:

If the structure of Word 1 is already included in Word 3 or Word 3 and Word 4 combined, where Word 1 has been misused, then Word 1 can be dropped from our dictionary.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 2, Word 3, Word 4, etc.

 Copyright 2023 *The Speak Logic Project*

Because of the misuse of *Word 1*, we simply remove it from our dictionary which further reduces the size of our dictionary. The way to look at it, the structure of *Word 1* is already included in *Word 2* or in a combination of *Word 3* and *Word 4*, then we drop *Word 1* from our dictionary and use *Word 2* or the combination of *Word 3* and *Word 4* instead.

Example Number 6

If Entity 1 exits where it is referred by Word 1, then Word 2 cannot be used to refer to Entity 1. While that may or may not reduce our dictionary, it is good to know. However, if *Word 2* only exists to refer to *Entity 1*, then *Word 2* can be removed from our dictionary.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 1, Word 3, Word 4, etc.

From the table above, since *Entity 1* is already referred by *Word 1*, by removing *Word 2* from our dictionary, that further reduce the size of our dictionary.

Example Number 7

If a word exists and it does not identify an actual entity, then that word can be removed from our dictionary. Another way to look at it, *if Word 2 exits in our dictionary, where Word 2 does not point to Entity 2, then it does not exist.* We can simply cross it out, because there is no entity pointed to it. The way to look at it, since when we hear a word we think about an entity, if the entity does not exist ,we have nothing to think about when we hear that word. In this case, that word is equal to nothing. So, there is no need to have that word in our dictionary.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 1, Word 3, Word 4, etc.

As shown from the table above, since existence of *word 2* in our dictionary serves no purpose, because it does not point to any entity or identify any entity, removing it from our dictionary further reduces the size of our dictionary.

Example Number 8

If the structure of a word does not exist or the structure of a word does not exist in another word or in combination of other words, that word can be removed in our dictionary. In other words, if the structure of a word does not exist and it does not exist in another word or in a combination of words, then that word does not exist in our dictionary.

Copyright 2023 *The Speak Logic Project*

In our dictionary below, to the left we have *Word 3*. The structure does not exist and it does not exist in any word in *Word 1*, in any other words, in any combination of *Word 1* and *Word 2* or any other combination. Then, *Word 3* can be removed from our dictionary.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 1, Word 2, Word 4, etc.

From the table above, by removing *Word 3* from our dictionary, that further reduce the size of our dictionary. The way to look at it, *Word 3* must point to an entity where that entity is being viewed as a structure. Since that structure does not exist, then it is not possible to use a word or a combination of words to identify that structure. Therefore, removing a word from our dictionary, since it does not identify anything, further reduces the size of our dictionary.

Example Number 9

If the usage of word develop problems or may develop problems, where the solution is another word or the opposite word, that word can be removed or replaced by the opposite word.

The way to look at it, *Word 1* points to *Entity 1*, where *Entity 1* develops problems, or its interpretation develops problems or *Entity 1* does not solve problems but causes them. Now, since *Word 2* points to *Entity 2*, where *Entity 2* is viewed as the solution for *Entity 1* or the problem caused by *Entity 1*, or *Entity 2* is the opposite of *Entity 1* or *Word 2* is the opposite of *Word 1*, then *Word 2* can be substituted for *Word 1*. Since *Word 1* develops problems rather than solving them, then *Word 2* can be replaced in our dictionary where *Word 1* is removed.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 2, Word 3, Word 4, etc.

As shown from the table above, by removing *Word 1* from our dictionary, that further reduces the size of our dictionary. The way to look at it, the usage of a word from our dictionary develops problems, where the usage of the opposite word is the solution of that problems. Since the opposite word solves the underlying problem, then the opposite word already replaced the word that develops problems. In this case, there is no need to have it in our dictionary. Removing it from our dictionary further reduces the size of our dictionary.

Example Number 10

If Word 1 exists, where Word 1 was intended to solve Problem 1, but failed; if Problem 1 solution exists in Word 2 or in combination or the structure of another in combination, then Word 1 can be dropped from our dictionary.

 Copyright 2023 *The Speak Logic Project*

The way to look at it, *Word 1* points to *Entity 1*, but *Entity 1* is not the solution for *Problem 1*. However, *Problem 1* solution exists in the form of: *Word 2* points to *Entity 2* or in combination of *Word 2* and *Word 3*. In this case, *Entity 1* is not needed as well as *Word 1*.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 2, Word 3, Word 4, etc.

From the table above, by removing *Word 1* from our dictionary, that further reduces the size of our dictionary. The way to look at it, a word points to an entity that is intended to solve a problem but failed. In other words, a word exists to identify the solution of a problem, but that solution does not exist because it failed. Since the entity that word points to does not exist, that words does not exist as well. Another entity exists as the solution for that problem and it is identified in the dictionary as *Word 2* or a combination of words. Since the word that points to an entity that does not exist in our dictionary serves no purpose, then it can be removed from our dictionary. In this case, the word that identifies the solution or the combination can be used instead. Doing so further reduces the size of our dictionary.

Example Number 11

If a word exists in the dictionary and it is misinterpreted and misunderstood, it can then be replaced by an existing word that is well understood and well interpreted. The way to look at it, if $Word\ 1 \xrightarrow{\text{point to}} Entity\ 1$ and $Word\ 2 \xrightarrow{\text{point to}} Entity\ 1$, and *Word 2* has been misunderstood or misinterpreted, then it is better to use *Word 1* in our communication rather than *Word 2*, because it makes our communication more portable and understood. Here we use the arrow to show words point to entities.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 1, Word 3, Word 4, etc.

As shown from the table above, since both words point to the same entity, they both identify the same entity. Since *Word 1* makes our communication portable and much more understandable, and *Word 2* adds ambiguity in our communication, it is better to drop it from our dictionary and keep *Word 1*. Doing so further reduces the size of our dictionary.

Example Number 12

If a word exists in our dictionary and it identifies the wrong entity, it can then be removed from our dictionary. This is the way to look at it, if *Word 1* exists, where

 Copyright 2023 *The Speak Logic Project*

Word 1 $\xrightarrow{\text{point to}}$ *Entity* 1. There also exists *Word 3* which does not point to *Entity 1*. In this case we can see that *Word 3* points to the wrong entity. This word does not identify a proper entity from our dictionary.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 1, Word 2, Word 4, etc.

As shown from the table above, a word in our dictionary does not identify the right entity. That word points to the wrong entity. Since we already have an existing word in our dictionary that identifies the right entity, there is no need to keep the word that identifies the wrong entity. Keeping that word in our dictionary takes more space and it makes our dictionary less portable. In this case, we simply remove it from our dictionary so we can reduce its size to match our parent's dictionary.

Example Number 13

If a word refers to a group where that word has been misunderstood, use the actual word in the group rather than the word itself. To better understand, let's take it like this. *Word 3* is viewed as a group of words where it includes both *Word 1* and *Word 2*.

Since *Word 3* has been misunderstood, it is better to use *Word 1* or a combination of *Word 1* and *Word 2* instead.

Before	After
Word 1, Word 2, Word 3, Word 4, etc.	Word 1, Word 2, Word 4, etc.

From the table above, since the word is not understood, it is better for us to refer to the actual entity which is identified by another word or group of words. Since that word and the group of words are already in our dictionary, we can use them instead and drop the misunderstood word from our dictionary. That will further reduce the size of our dictionary.

Example Number 14

Here, let's take another look at relationship with Visual Aspect of Communication. The term visual aspect of communication means that we communicate relatively about entities that we identify. Those entities do not have to be visually identified. As long as we can identify them, then they enable our communication.

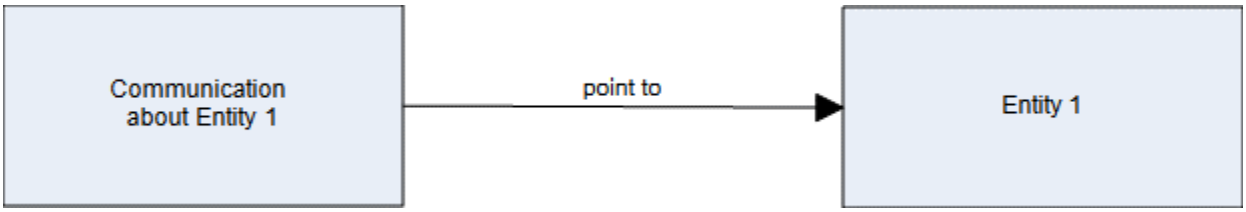
Visual Aspect of Communication enables us to communicate about entities that we identify. For example, if we identify *Entity 1*, then *Entity 1* enables our communication so we can

 Copyright 2023 *The Speak Logic Project*

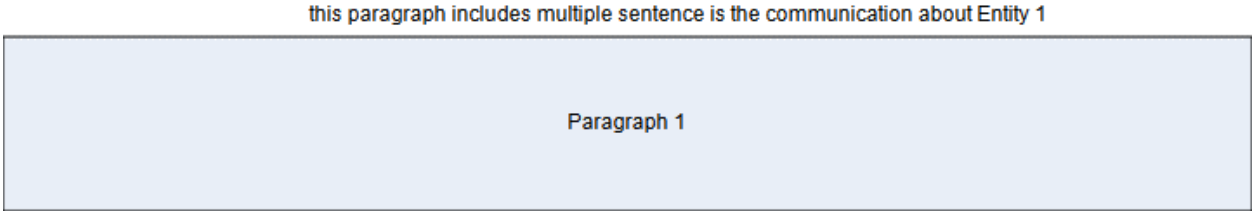
communicate about *Entity 1*. At the same time, if we don't identify *Entity 2*, then *Entity 2* does not enable our communication. In this case, it is not possible for us to communicate about *Entity 2*. From that we can see an entity that we identify enables our communication while an entity that we don't identify does not enable our communication. Based on that, we can see the communication of *Entity 1* exists because of the existence of *Entity 1*, while the communication of *Entity 2* does not exist because *Entity 2* does not exist.

In other words, visual aspect of communication enables us to communicate about entities that we identify. It does not matter if that entity is physical or not. As long as we can identify it, then it enables our communication. In this case, it does not matter if we can identify the entity with our eyes or not. As long as we identify it, it enables our communication. In other words, if an entity exists, and we can identify that entity, then that entity enables our communication.

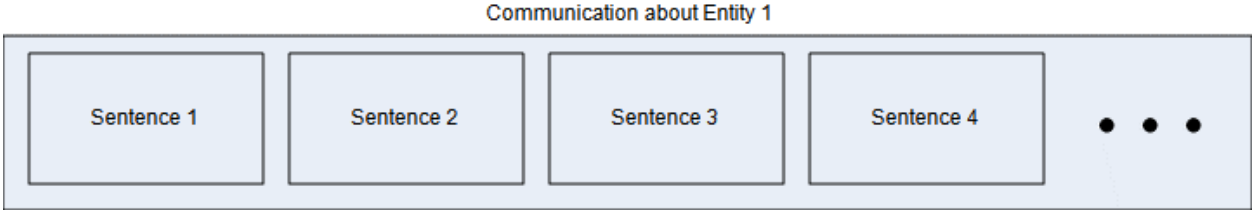
To better understand the explanation above, we can think it like this. The existence of *Entity 1* enables the communication about *Entity 1* to point to it in this form.



Here, if we want to in terms of oral and written communication, we can identify that communication as a group of sentences in the form below, which we simply call a paragraph.

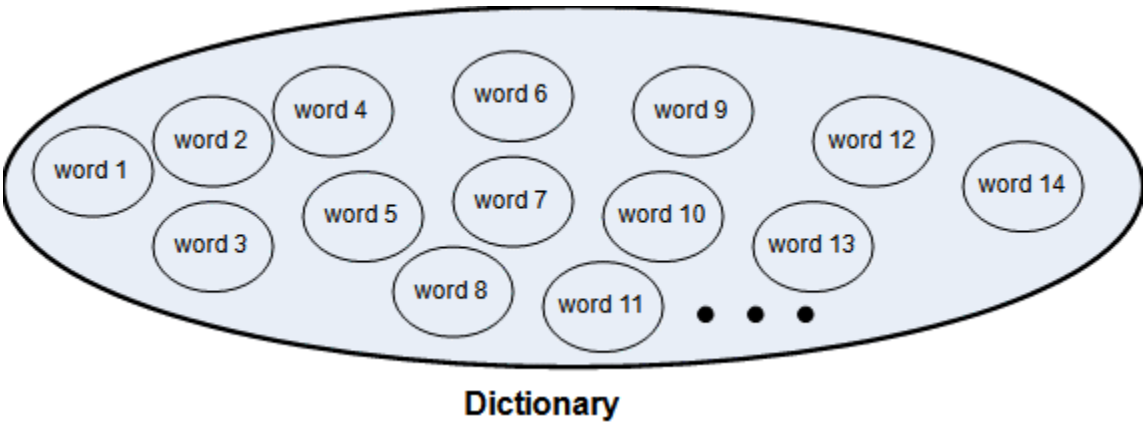


Since a sentence is a group of words, each sentence in the paragraph above can be viewed as a group of words so does the communication about the entity. In this case, we can represent that communication in the form below.



From above, it shows that oral communication about *Entity 1* which is viewed as a collection of word is made of a group of sentences where each sentence is viewed as a group of words. Now,

since all words in that communication belong to the dictionary, we can represent that in the form below.



In other words, all words that make up communication about *Entity 1* belongs to the dictionary. What is important here is that the fact that *Entity 1* exists, all words related to *Entity 1* exist in the dictionary. The fact That *Entity 1* exists, all words that make up communication about *Entity 1* also exist in the dictionary. By understanding that, we can see since *Entity 2* does not exist, its communication does not exist and words that make up that communication do not exist and they are not in the dictionary. By understanding that, we can see since words that make up communication about *Entity 2* do not exist, they can then be removed from our dictionary as shown by the table below. For example, let's assume that *Word 18* to *Word 22* make communication about *Entity 2*.

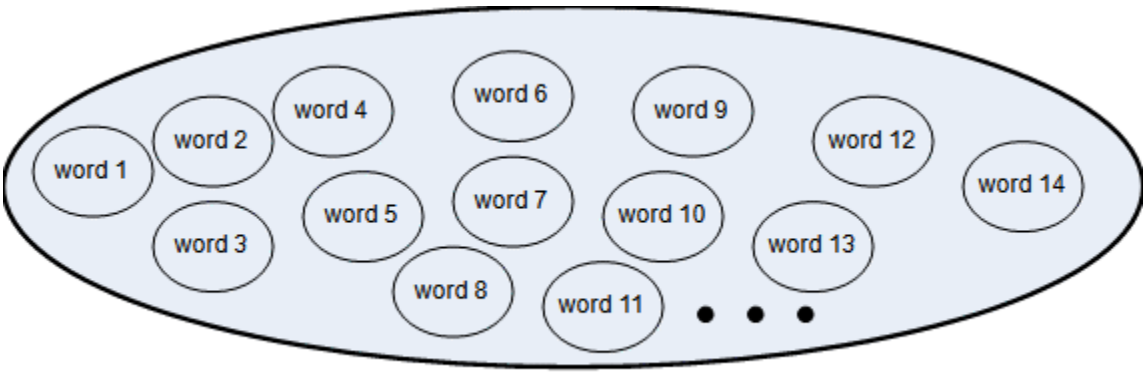
Before	After
Word 1 to Word 17, Word 18 to Word 22, Word 23, etc.	Word 1 to Word 17, Word 23, etc.

From the table above, it shows our dictionary contains two sets of words: a set for entities that are identified or can be identified and a set for entities that are not identified or cannot be identified. Since the communication of the set of words that does not identify entities is not enabled, then it is simply removed from our dictionary. By removing them from our dictionary, that further reduces the size of our dictionary and makes our communication more portable. By understanding that, we can see *if a word is not identified by an entity or cannot be identified by an entity, then that word should be removed from our dictionary*. We can also say that *if a word is not identified by an entity, then the communication using that word is not enabled*. This is the same as saying that *if a word does not identify an entity, then that word cannot be used in communication*.

 Copyright 2023 *The Speak Logic Project*

To better understand the explanation from the paragraph above, we can think it like this. If a word exists in a dictionary, then it points to an actual entity. While some of us may not understand or be familiar with that entity, when we hear that word, we can ask questions and find information about it that word points to the entity that it identifies. Now, if the entity does not exist, then the information about the word does not exist and it is not possible for us to think properly about it. Since that word points to no entity, leaving it in our dictionary serves us no purpose. It simply takes space in our dictionary that simply increases the size of it. In this case, removing it is the right thing to do and makes the dictionary more compact and portable.

Since existing entities have aspects, if an entity does not exist neither does its aspect. In this case, any words that refer to the aspect of that entity do not exist. To better understand what we have just said, let's take it like this. Assume that our dictionary contains the following words:



Dictionary

Let's assume that *Word 5* does not exist, neither does the aspect of *Word 5*. If *Word 3* and *Word 4* exist in our dictionary for the purpose of *Word 5* only, where they are viewed as aspects of *Word 5*, since *Word 5* cannot be identified or does not exist, neither can *Word 3* and *Word 4*. In this case, not only *Word 5* needs to be removed from our dictionary, but also *Word 3* and *Word 4* as shown below.

Before	After
Word 1, Word 2, Word 3, Word 4, Word 5, Word 6, Word 7, Word 8, etc.	Word 1, Word 2, Word 6, Word 7, Word 8, etc.

The way to look at it, the aspect of an entity enables its existence. If an entity does not exist, neither does the aspect of that entity. In this case, a word that refers to that entity does not exist as well. Since a word that identifies that entity does not exist, neither do words that identify aspects of that entity. By removing words that identify aspects of words that do not exist in our dictionary, we further reduce the size of our dictionary.

Copyright 2023 *The Speak Logic Project*

By taking a quick look at the feedback model from our parent, we can see that our parent takes visual aspect of communication into considered when providing feedback to use to enable the correction of our communication. In this case, we can see that if we repeat a word that points to an entity that does not exist or does not identify an actual entity, then our parent provides us feedback to use a word that identifies an actual entity instead. From that, we can see our parent takes visual aspect of communication during that process. By using words that identify actual entities, it makes our communication portable where others can understand it by identifying the actual entities those words refer to.

Example Number 15

This example is simply a summary of all the examples above. Here we are going to provide a practical example using all the techniques provided above.

We need to reduce the size of our dictionary so it matches our parent's dictionary. We need to do that so we can be self-sufficient where we don't have to depend on our parent to help us correct our communication. By doing so, we can correct our own communication. Here as an example, we have our dictionary has a dictionary that contains 70 words, where 20 of them are bad or negative words. In this case, we represent our dictionary below to show both the group of good words and the group of bad words.

Our Dictionary	
Good Words	**Bad Words**
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 50	Word 51, word 52, word 53, word 54, word 55, word 56, word 57, word 58, ⋯, word 70

The table above represents our dictionary which contains both good words and bad words. We group both the good words and bad word to represent them. We use the triple periods to show continuity of words because of insufficient space. From the table above, *Word 1* can also be identified as *Good Word 1*, while *Word 51* can also be identified as *Bad Word 51*. Since we title them as both good words and bad words, that is sufficient.

Since we no longer need the set of bad words in our dictionary, we simply eliminate them by deleting them from our dictionary. In this case, our dictionary is represented below with only the set of good words.

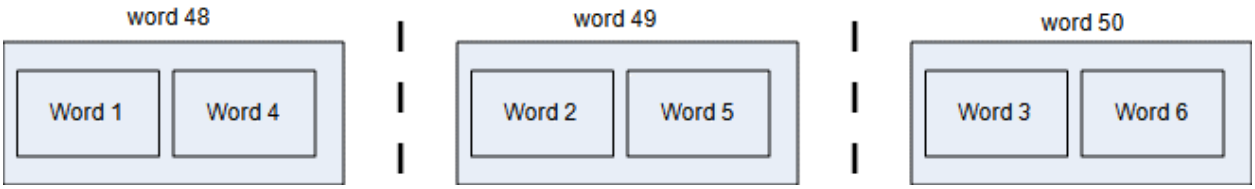
Our Dictionary
Good Words
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 50

Copyright 2023 *The Speak Logic Project*

Since now we only have good words in our dictionary, we can simply represent our dictionary above without using the good words title.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 50

Now, let's apply the technique from example **number 2** to further reduce the size of our dictionary. From example **number 2**, we have learned that *if two or more words can be combined to form another word or a new word, the new word or the other word may not be needed*. In this case, let's assume that we have 3 words that can be combined with another 3 words to form 3 new words. For instance, we have *Word 1*, *Word 2*, and *Word 3* that can be combined with *Word 4*, *Word 5*, and *Word 6* to form *Word 48*, *Word 49*, and *Word 50*. We can represent them in the form below:



By using this approach, *Word 48*, *Word 49*, and *Word 50* are no longer need in in our dictionary. By eliminating them from our dictionary, we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 47

This approach further reduces the size of our dictionary. As we can see, we start with 70 words, now we are down to 47.

Now, let's apply the approach from example **number 3**. From the approach number 3, we have *learned that if a word exists in our dictionary and it is confused or caused a lot of confusion, it can then be replaced by another word*. From our dictionary, we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 47

Let's assume that the usage of *Word 46* and *Word 47* have been caused a lot of confusions, they can then be removed from our dictionary. By removing them, we further reduced the size of our dictionary as shown below.

 Copyright 2023 *The Speak Logic Project*

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 45

The way to look at it, when we use those two words, they make people confused or create ambiguity in our communication. Since we already have words in our dictionary that can be used to remove that confusion, we simply remove those two words and use the existing words. Doing so makes our communication more portable and further reduces the size of our dictionary.

Another way to look at it, since communication about *Entity 1* points to *Entity 1*. It can be viewed as a group of sentences which is a collection of words, where all those words belong to the dictionary. Now, since *Word 46* and *Word 47* create confusion in the communication about *Entity 1*, using those two words does not enable communication about *Entity 1* to point to *Entity 1*. In this case, those two words can simply be removed from communication about *Entity 1*. When we do that, we simply remove them from our dictionary which makes it more portable.

From the approach from example **number 4**, we have learned that. *If a word or the definition of a word is already defined or implemented in another word or a group of words, then the usage of that word can be dropped*.

From our dictionary, we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 45

Let's assume that *Word 43*, *Word 44*, and *Word 45* are already defined in other words, where those words have been misused. But when the other words are used, there is no problem. In this case, we can simply remove them from our dictionary to prevent the misused. Removing them from our dictionary prevents the misused. It also reduces the size of our dictionary and makes our communication more portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 42

Now, let's use the approach from example **number 5**. *If the structure of a word is already included in another word or a combination of words, then that word can be dropped; especially if that word has been misused, misinterpreted or misunderstood.*

Our Dictionary

Copyright 2023 *The Speak Logic Project*

Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 42

From the approach above, let's assume that the structure of *Word 41* and *Word 42* is already implemented in *Word 2* and *Word 3* where *Word 41* and *Word 42* have been misused. In this case, we can simply drop them in our dictionary. By dropping them from our dictionary, we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 40

As shown above, since the two words have been misused and the structures of both words are already implemented in other words, there is no need to use them since they have been misused. In this case, we can simply remove them from our dictionary and use the existing words that have not been misused. Doing so reduces the size of our dictionary and also makes it more portable.

Now, let's look at the approach of example **number 7**: *If a word exists and does not identify an actual entity, then that word can be removed from our dictionary*. From our dictionary we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 40

Let's assume that *Word 38*, *Word 39* , and *Word 40* exist in our dictionary but do not point or identify any entity. For example, *Word 38* does not identify *Entity 38*, *Word 39* does not identify *Entity 39* and *Word 40* does not identify *Entity 40*.

As explained above, those words do not point to actual entities. In this case, we can simply remove them from our dictionary because they serve us no purpose. They are simply using more space in our dictionary to make it less portable. By removing them, our dictionary becomes more compact and more portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 37

Now, let's take a look at the approach in example **number 8**. *If the structure of a word does not exist or does not exist in another word or combination of other words, that word can be removed from our dictionary*.

Our Dictionary

 Copyright 2023 *The Speak Logic Project*

Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 37

Let's assume that from our dictionary above, we have *Word 36* and *Word 37*. Those two words should point to two entities. The fact that the structures of those two words do not exist or do not exist in a combination of words, they simply serve no purpose in our dictionary. We can also say that those words have no structure, because they don't point to any entity. They serve us no purpose because we cannot use them in our communication or communicate properly with them since they identify no entity. In this case, we can simply drop them from our dictionary. By removing them from our dictionary, that makes our dictionary more compact and portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 35

Let's take a look at the approach from example **number 9**. *If the usage of word develops problems or may develop problems, where the solution is another word or the opposite word, that word can be removed or replaced by the opposite word.*

We have the following words in our dictionary :

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 35

If the usage of *Word 35* develops problems or *Word 34* is viewed as the solution for that problem. In this case, *Word 35* can be dropped from our dictionary and use *Word 34* instead, which is the solution of the problem. The way to look at it, a problem is developed because of the usage of a word; the opposite word which is viewed as the solution can be used instead. For instance, *Word 34* points to *Entity 34* where *Entity 34* is viewed as the solution entity for the identified problem. Now, *Word 35* points to *Entity 35*, but *Entity 35* only exists when *Word 35* is used. In this case, it makes sense for us to drop *Word 35* from our dictionary to prevent the development of that problem. In other words, the problem only exists when the word is used. To solve that problem, we can simply drop that word from our dictionary and use the solution word instead. When we do that, we further reduce the size of our dictionary and make our communication more portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 34

Now, let's look at the approach of example **number 10**. *If a word exists where that word was intended to solve a problem, then it failed. If that problem solution exists in another word or in*

 Copyright 2023 *The Speak Logic Project*

combination of words or structure in other word(s), then that word can be dropped from our dictionary.

In our dictionary we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 34

If *Word 34* exists in our dictionary where it was intended to solve a problem, then it failed. If that problem solution exists in *Word 33* or in combination of *Word 32* and *Word 33* or in the structure of any of them, then we can drop *Word 34* and use *Word 33* or the combination of *Word 32* and *Word 33*. The way to look at it, if that word is supposed to be the solution of a problem, but failed to solve that problem, then another word that is viewed as the solution for the actual problem can be used instead. In this case, we can simply drop that word from our dictionary. Another way to look at it, a problem is identified where *Word 34* is supposed to point to *Entity 34* which is viewed as the solution for the problem. But *Entity 34* is not the actual solution for that problem, because it fails to solve it, while *Word 33* exists in the form of *Word 33* points to *Entity 33* where *Entity 33* is viewed as the solution for the actual problem. In this case, there is no need in our dictionary for *Word 34*. Another way to look at it,

$$Word\ 33 \xrightarrow{point\ to} Entity\ 33$$

$$Word\ 34 \xrightarrow{not\ point\ to} Entity\ 33$$

In this case, we simply remove *Word 34* from our dictionary, because it does not point to an actual entity. Now since the word that intended to solve the problem fails, it does not point to an actual entity or an actual solution. The actual word that solves the problem is the one that points to the actual solution for that problem. In this case, the intended word can be removed from our dictionary. When we do that, we further reduce the size of our dictionary and make our communication more portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 33

Let's look at the approach of example **number 11**. *If a word exists in the dictionary and it is misinterpreted and misunderstood, it can then be replaced by an existing word that is well understood and well interpreted*.

Assume that in our dictionary we have:

Our Dictionary

 Copyright 2023 *The Speak Logic Project*

Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 33

Now, if *Word 32* and *Word 33* have been misinterpreted and misunderstood where *Word 30* and *Word 31* are well understood and well interpreted, then we can drop *Word 32* and $\hat{w}_{33}$ in our dictionary and use *Word 30* and *Word 31*. The way to look at it:

Word 30 $\xrightarrow{\text{point to}}$ *Entity* 30

Word 31 $\xrightarrow{\text{point to}}$ *Entity* 31

But when we use $\hat{w}_{32}$ and $\hat{w}_{33}$ in our communication, they are viewed in the form of:

Word 32 $\xrightarrow{\text{does not point to}}$ *Entity* 30

Word 33 $\xrightarrow{\text{does not point to}}$ *Entity* 31

The way to look at it, those words supposed to point to the same entities. But when they are used, they don't point to the same entities, because they have been misunderstood and misinterpreted. Therefore, they can be dropped from our dictionary and use the existing words in our dictionary that are well understood and well interpreted. As pointed out above, those words still point to the same entity. When we do that, we further reduce the size of our dictionary and make our communication more portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 31

From the approach of example **number 12**, we have: *If a word exists in our dictionary and it identifies wrong entity, it can then be removed from our dictionary*.

From our dictionary we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 31

Let's assume that *Word 31* identifies wrong entity, in this case we have

Word 30 $\xrightarrow{\text{point to}}$ *Entity* 30 but *Word* 31 $\xrightarrow{\text{wrongly identified}}$ *Entity* 30

From above, it shows that *Word 30* identifies *Entity 30*, but *Word 31* wrongly identifies *Entity 30*. This is the same as saying that an actual word points to an actual entity. Another word

 Copyright 2023 *The Speak Logic Project*

wrongly points to the actual entity. In this case, we can remove the other word from our dictionary since we already have a word that points to the actual entity. When we do that, we further reduce the size of our dictionary and make our communication more portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 30

Now, let's take a look at the approach of example **number 13**. *If a word refers to a group where that word has been misunderstood, use the actual word in the group rather than the word itself.*

In our dictionary we have:

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 30

If *Word 30* refers to a group of word and *Word 30* has been misunderstood or misinterpreted, we can use a specific word in the group rather than using *Word 30*. Now, if *Word 30* only exists in our dictionary for that purpose, we can simply drop it from our dictionary. When we do that, that further reduces the size of our dictionary and make our communication more portable.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 29

Now, we have reduced the size of our dictionary by applying all the approaches identified from the examples. After doing so, we have reduced the size of our dictionary to 29 as shown in the table above.

Now, we have only 29 words in our dictionary. Let's assume that after applying all the techniques above, we have identified 4 words in our dictionary that do not identify actual entities. In other words, using visual aspect of communication, there are 4 words in our dictionary that do not identify any entity. Those 4 words are: *Word 26*, *Word 27*, *Word 28*, and *Word 29*. The way to look at it, the 4 words identified here are in the dictionary, but they do not identify any entity, for example we have:

$$\textit{Word } 25 \xrightarrow{\text{point to}} \textit{Entity } 25$$
$$\textit{Word } 26 \xrightarrow{\text{not point to}} \textit{Entity } 26$$
$$\textit{Word } 27 \xrightarrow{\text{not point to}} \textit{Entity } 27$$
$$\textit{Word } 28 \xrightarrow{\text{not point to}} \textit{Entity } 28$$
$$\textit{Word } 29 \xrightarrow{\text{not point to}} \textit{Entity } 29$$

 Copyright 2023 *The Speak Logic Project*

From above, we can see that *Word 26, Word 27, Word 28,* and *Word 29* point nowhere. While the other words in our dictionary point to actual entities, the 4 words identified do not point to actual entities. They simply occupy more space in our dictionary. By removing them from our dictionary, we simply shrink the size of our dictionary to make it more compact and portable. Removing them from our dictionary simply reduces the size of our dictionary to just 25 words as shown below.

Our Dictionary
Word 1, word 2, word 3, word 4, word 5, word 6, word 7, word 8, ⋯, word 25

As we can see, we start at 70 words. By removing all negative from our dictionary, we reduce it to 50 words. By applying all the techniques identified, we further reduce our dictionary to 25 words. In this case, we decrease the size of our dictionary to about a third.

Some other approaches that are good to know which may not further reduce the size of our dictionary, but will improve our communication. If *Word 1* and *Word 2* does not point to *Entity 1*, then we use *Word 1* to communicate about *Entity 1* instead of *Entity 2*.

If the combination of *Word 1* and *Word 2* do not point to an entity, then we do not use it in our communication. This is the same as saying that *if a term or an expression exists and it does not point to an actual entity, then we cannot use it in our communication*.

Example Number 16

This should have already been taken care by visual aspect of communication. This example only gives additional explanation. *If Word 1 exist in our dictionary where Word 1 does not promote stability, then Word 1 can be dropped from our dictionary*. To better understand, let's take it like this. A word exists where that word points to an entity. The existence of that entity enables the existence of that word. Since information exists before our communication, the usage of that word enables us to provide information about the entity that word points to. In terms of providing information, we can think it like this. I am communicating with you, where I use *Word 1* to provide information to you. In this case, your state must be preserved or become much better after the information. In other words, after using that word, you should not feel abnormal. After using that word to communicate, one should not feel abnormal. If that word does not preserve our stability, then it can be dropped from our dictionary.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 2, word 3, word 4, word 5, ⋯

From the table above, *Word 1* is viewed as a word in our dictionary where its usage promote

 Copyright 2023 *The Speak Logic Project*

instability or make people feel abnormal. In this case, we can drop *Word 1* from our dictionary. When we do that, we further reduce the size of our dictionary. The way to look at it, by dropping *Word 1* and using an existing word in our dictionary that promotes stability, we improve our communication by making our communication more portable and we also reduce the size of our dictionary.

In terms of visual aspect of communication, usually if *Word 1* exists in our dictionary for the sole purpose of promoting instability or making people feel abnormal, it does not point to an actual entity. The way to look at it is that *Word 2* points to *Entity 2*, now when we use *Word 1* in our communication, we try to make it point to an entity, where an actual word that already exists in our dictionary already points to that entity. For example *Word* 1 $\xrightarrow{\text{does not point to}}$ *Entity* 2 or *Word* 1 $\xrightarrow{\text{tries to point to}}$ *Entity* 2; since that word does not point to the actual entity and we try to make it point to the actual entity, it is better for us to drop that word from our dictionary and use the actual word that points to the actual entity. In this case, we can say that we drop that word from our dictionary and use the existing word that actually identifies the entity instead.

Example Number 17

This example is the same as visual aspect of communication. This example simply provides additional explanation of visual aspect of communication. *If Word 1 exists in our dictionary where Word 1is unique and cannot be compared, if Word 2 exists in our dictionary solely to compare Word 1, then Word 2 can be dropped from our dictionary*. In other words, *if a word exists in our dictionary where that word is not comparable, another word cannot exist in that dictionary for solely purpose to compare to that word*.

To better understand the approach above, let's take it like this. *Entity 1* exists in a form where it is unique to itself. In this case, since the existence of an entity enables the existence of word that point to it, then we have *Word* 1 $\xrightarrow{\text{points to}}$ *Entity* 1. Because only a single *Entity 1*exists and information about *Entity 1* points to *Entity 1*, it may not be possible to find another entity to compare *Entity 1* with. In this case, a word that exists in our dictionary solely to compare *Entity 1* does not point to an actual entity or identify an actual entity. In this case, that word can be dropped from our dictionary.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 1, word 3, word 4, word 5, ⋯

From the table above, *Word 1* exists in *our dictionary* and *Word 1* is not comparable. Since *Word 2* exists in our dictionary only to compare with *Word 1*, then the fact that *Word 1* is not

 Copyright 2023 *The Speak Logic Project*

comparable, then *Word 2* is dropped from our dictionary. When we do that, we further reduce the size of our dictionary and make our communication more portable. From visual aspect of communication, we can see that *Word* $1 \xrightarrow{\text{point to}}$ *Entity* 1, but *Word* $2 \xrightarrow{\text{not point to}}$ *Entity* 2 where *Entity 2* is given in a form that is comparable to *Entity 1* as well as *Word* $1 \xrightarrow{\text{not point to}}$ *Entity* 2. In this case we can see that *Entity 2* does not exist or exist in a form that is comparable to *Entity 1*, therefore it makes sense to drop *Word 2* from our dictionary because it does not point to an actual entity.

Another way to look at it is that the comparison of two entities depends on the entities not on us. Since communication exists after information, the information about comparison of two entities enables communication about that information. If that information does not exist, that communication does not exist as well. In other words, the comparison information of the entities enables the communication about that comparison. Since that information does not exist, that communication is not enabled by that information.

Example Number 18

This example is the same as visual aspect of communication. This example simply provides additional explanation of visual aspect of communication. *If Word 1 and/or Word 2 exists in our dictionary, where they refer to Word 3 where Word* $3 \xrightarrow{\text{point to}}$ *Entity* 3*, but Entity 1 and/or Entity 2 do not exist. In this case, Word 1 and/or Word 2 can be dropped from our dictionary*. The way to look at it, a word exists in our dictionary if it refers to an entity. In other words, if it is already points to that entity, then that word can be dropped. Especially if that word has been misused or misinterpreted or misunderstood.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 3, word 4, word 5, word 6, word 7, ⋯

From the table above, we remove the two words from our dictionary, because we already have an existing word that points to an actual entity. In terms of visual aspect of communication, *Word* $1 \xrightarrow{\text{not point to}}$ *Entity* 1 and *Word* $2 \xrightarrow{\text{not point to}}$ *Entity* 2 but when *Word 1* and/or *Word 2* are used they refer to *Word* $3 \xrightarrow{\text{point to}}$ *Entity* 3. In this case, we can drop *Word 1* and/or *Word 2* and use $\hat{w}_3$ instead. When we do that, we further reduce the size of our dictionary.

Example Number 19

Copyright 2023 *The Speak Logic Project*

This example is the same as visual aspect of communication. This example simply provides additional explanation of visual aspect of communication. *If Word 1 does not exist,* *Word* 1 $\xrightarrow{\text{not point to}}$ *Entity* 1. *In this case, if Entity 2 is referred to as the function of Entity 1 and Entity 2 does not exist, then Word 2 which does not point to Entity 2 in the form of Word* 2 $\xrightarrow{\text{not point to}}$ *Entity* 2 *does not exist as well*. In other words, *if an entity does not exist or it cannot be identified, then function of that entity does not exist as well*. Any word that refers to that entity and function of that entity does not exist and can be dropped from our dictionary.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 3, word 4, word 5, word 6, word 7, ⋯

From the table above, since *Word 1* does not exist and *Word 2* which refers to the function of *Word 1* does not exist as well, in this case, we remove both of them from our dictionary to further reduce the size of our dictionary.

Example Number 20

This example is the same as visual aspect of communication. This example simply provides additional explanation of visual aspect of communication. *If Word 1 does not exist, then the combination of Word 1 with Word 2 does not exist as well.* In other words, *if a word does not exist, then the combination of that word does not exist*. This is the same as saying that: *If a word does not exist, the combination of that word with another/other word(s) does not exist as well*. For example, if *Word 1* exists then *Word* 1 $\xrightarrow{\text{point to}}$ *Entity* 1. If *Word 2 does not* exist, then *Word* 2 $\xrightarrow{\text{not point to}}$ *Entity* 2. In this case, it is not possible to have a combination of *Word 1* and *Word 2* because *Entity 2* does not exist.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 1, word 3, word 4, word 5, word 6, ⋯

The way to look at it, in terms of visual aspect of communication, we can identify *Entity 1* but we cannot identify *Entity 2*. In this case, the combination of *Word 1* and *Word 2* does not identify an actual entity. That should have further reduced the size of our dictionary from the previous example. Here it is only provided as more information for terms and expressions, which makes our communication more portable.

Example Number 21

 Copyright 2023 *The Speak Logic Project*

This example is the same as visual aspect of communication. This example simply provides additional explanation of visual aspect of communication. *If Entity 1 and Entity 2 exist, where Entity 3 exists combination of Entity 1 and Entity 2, then Word 3 can be used to identify Entity 3 in the form of Word* $3 \xrightarrow{\text{point to}} \textit{Entity}\ 3$. However, *If Entity 1 and Entity 2 exist, where Entity 3 does not exist in combination of Entity 1 and Entity 2, then Word 3 cannot be used to identify Entity 3, because either Entity 3 does not exist or it does not exist in combination of Entity 1 and Entity 2*. In this case, we can simply drop *Word 3* from our dictionary which further reduces the size of our dictionary.

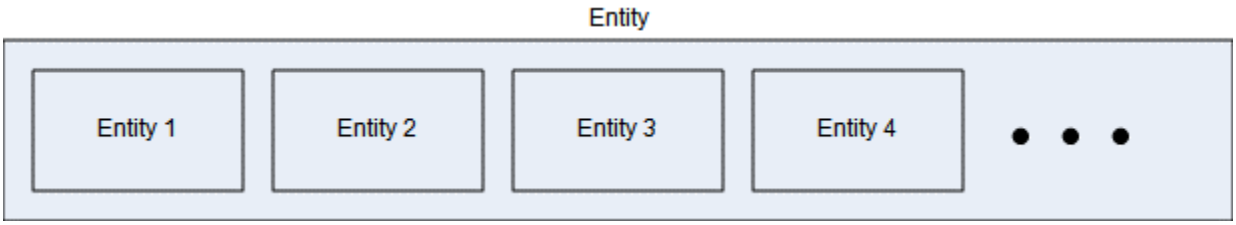
In other words, *if two entities exist where another entity is the result of that combination, then a word can be used to identify that entity*. However, *if two entities exist where another entity exists and it is not the result of the two entities, then another word cannot be used to identify that entity*. By removing that word from our dictionary, we further reduce the size of our dictionary.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 1, word 2, word 3, word 5, word 6, ⋯

From the table above, *Word 3* exists as a combination of *Word 1* and *Word 3* which points to *Entity 3*. However , *Word 4* does not exist in combination of *Word 1* and *Word 2* or simply *Word* $4 \xrightarrow{\text{not point to}} \textit{Entity}\ 4$. In this case, *Word 4* can be removed from our dictionary because it does not identify an actual entity. Another way to look at it, if two words can be combined to form a new term or expression, however if one word does not exist, that term or expression does not point to an actual entity. If that term or expression or word exists in our dictionary, by removing it from our dictionary, we reduce the size of our dictionary and make it more portable.

Example Number 22

This example is the same as visual aspect of communication. This example simply provides additional explanation of visual aspect of communication. *If an entity exists as a combination of multiple entities in the form of:*



 Copyright 2023 *The Speak Logic Project*

From the model above, the identified entity exists as a combination of multiple entities, where all of them point to that entity. In other words, all the entities inside combined, point to the outside entity.

If *Word 1* exists in our dictionary, where *Word 1* points to the outside entity, in this case, *Word 1* points to *Entity 1*, *Word 2* points to *Entity 2*, *Word 3* points *Entity 3*, and so forth.

However, *if Word 2* exists in our dictionary and it does not point to the outside entity in the form of *Word* $3 \xrightarrow{\text{not point to}} Entity$, then *Word 2* does not point to any entity inside the entity above. In this case, *Word 2* does not point to *Entity 1*, or *Entity 2*, and so forth.

If *Word 2* exists in our dictionary to show it points to the *Entity* although it does not, then it can be removed from our dictionary which further reduces the size of our dictionary. In other words, *if an entity exists in a form where that entity includes multiple parts that point to that entity; if a word exists in the dictionary where that word points to that entity, that word also points to other parts of that entity. However, if a word exists in the dictionary where that word does not point to that entity, that word does not point as well to other parts of that entity.* That word can then be removed from the dictionary which further reduces the size of the dictionary.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 1, word 3, word 4, word 5, word 6, ⋯

From the table above, *Entity 1* is presented in a form where multiple parts of that entity point to that entity, where *Word 1* points to that entity. Since *Word 2* exists in the dictionary and shows that it points to that entity or part of that entity, although it does not, we simply drop it from the dictionary which further reduces the size of the dictionary. This technique is also applied to each sub entity in the main entity above. For example, if *Entity 1* is made of *Entity 11, Entity 12*, and so forth, the same process is also applied.

Example Number 23

This example is the same as visual aspect of communication. This example simply provides additional explanation of visual aspect of communication. *If Entity 1 exists where it points to Entity 2*, if *Word 2 exists in our dictionary and shows that Entity 1 points to Entity 3*, then *Word 3 can be removed from the dictionary because Entity 1 does not point to Entity 3*. Removing *Word 3* from our dictionary further reduce the size of our dictionary.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 1, word 2, word 4, word 5, word 6, ⋯

Copyright 2023 *The Speak Logic Project*

In other words, *if an entity points to another entity and a word exists in our dictionary that shows that entity points to another entity rather than the actual entity it points to, then that word can be removed from our dictionary.* When we do that, we further reduces the size of our dictionary and make it more portable.

Example Number 24

From now, we already know that by substituting words in our dictionary or vocabulary by other existing words, that reduces the size or our dictionary. *If a word cannot solve a problem, where a substitute word can solve that problem, it is better to use the substitute word rather than the word*. The fact that the substitute word exists in our dictionary, the word that is replaced can be dropped. If *Word 5* exists in our dictionary, where *Word 5* cannot solve *Problem 1*, and *Word 2* exists where it can solve *Problem 1*, then we can drop *Word 5* from our dictionary and use *Word 2* instead. This further reduce the size of our dictionary as shown below.

Before	After
Word 1, word 2, word 3, word 4, word 5, ⋯	Word 1, word 2, word 3, word 4, word 6, ⋯

From the table above, the fact that the word that can solve the problem exists, any word that does not solve the problem or simply expands it or replaces it by more problems, is discarded or dropped from our dictionary. In this case, we replace the word with words that can solve the identify problem. This is the way to look at it. A problem is identified a word used to claim to solve that problem, even though it does not solve the actual problem. Since the solution of that problem exists, we simply use the word that points to the solution rather that the word. Another way to look at it, even when the word *solution* exists, it is better to use it than the actual word that does not solve the problem.

From above, we can see that if we identify a problem, rather than using words that do not solve it or words that claim to solve it, we can simply use words that solve it or simply the word *solution*. This process further reduces the size of our dictionary and prevents us from developing or creating more problems. In this case, we can ask these questions: Why have so many words in our dictionary if they don't solve any problems? Why have so many words in our dictionary if they don't help us solve any problems?

From the above, we can see that if we use the word that claims to solve the problem and it does not solve, the problem still remains. The underlying problem can also be expanded to other problems. In this case, using that word does not do us any good. Using that word does not help

 Copyright 2023 *The Speak Logic Project*

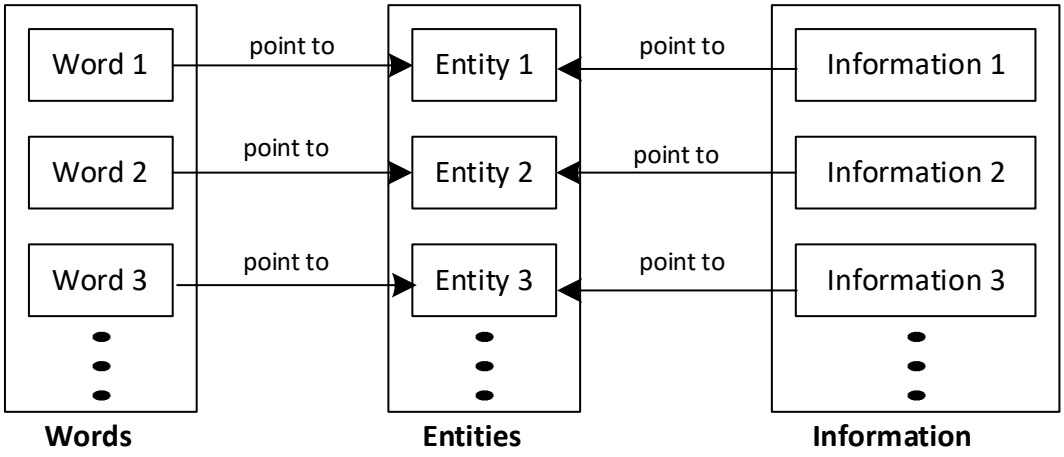
us solve the underlying problem. In this case, we can simply drop it and replace it with the word that helps us solve the problem or simply the word *solution*. As we can see, by dropping that word from our dictionary, that further reduces the size of our dictionary and we also solve problems. Leaving such words in our dictionary does not help us solve any problems. It simply increases the size of our dictionary and it is useless to us since it does not help us. This is why it is good to reduce the size of our dictionary.

The function of our application is a function of communication. If we are able to communicate to execute a function to solve an identified problem, then the collection of words usage for that communication is sufficient. In this case, no extra word is needed. From that, we can ask these questions: Why have so many words in our vocabulary if they don't help us communicate? Why have so many words in our dictionary if they don't help us in our communication? As we can see from the feedback process from our parent, we do not need so many words in our vocabulary or dictionary because having too many words does not help us at all in our communication. This is why we don't need too many words in our vocabulary or our dictionary.

Example Number 25

The overall techniques we have learned to reduce the size of our dictionary to be equivalent to our parent dictionary can be summarized as follow:

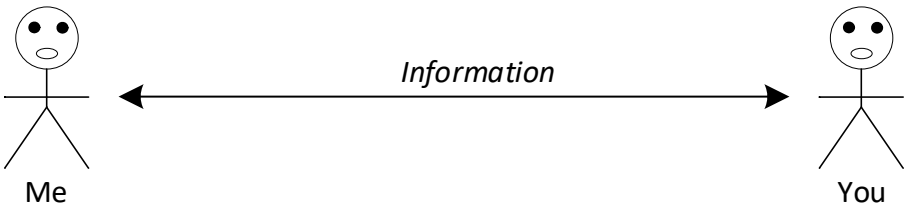
Since words point to entities and information about entities points to entities, then there is no need to have new or additional words, if new entities are not introduced. In other words, in our communication we use words to identify entities, where information about those entities point to them. In this case, if we do not add new entities, where new information does not point to new entities. There is no need to have new or additional words. To better understand the explanation, let's take it like this:



 Copyright 2023 *The Speak Logic Project*

From the model above, let's assume that we have a fixed set of entities with a fixed set of words; as well as information about the entities is fixed. Since no new entity is added, there is no need to have new or extra or additional words. When we do that, we simply manage our dictionary or vocabulary.

Another way to look at it, let's take it like this: communications between me and you where information flows between me and you as shown by this model.

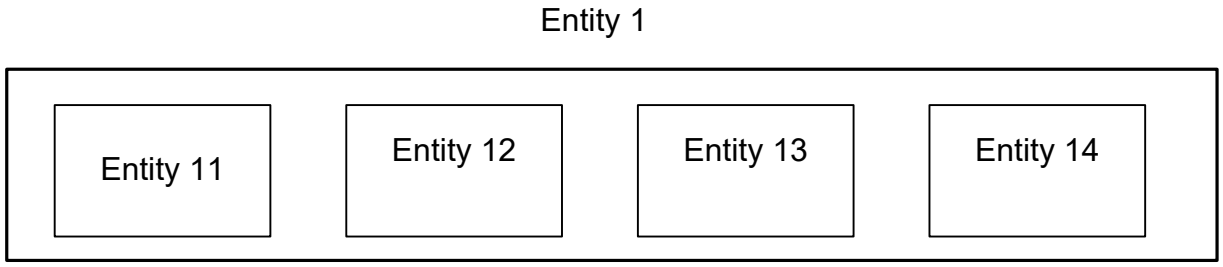


From the above, we can see that information flows in the communication interface. Since no new entities is introduced, no new word is added. So, the flow of information is limited by the existing entities. In this case, extra words are not needed.

Derived Words and Derived Entities

While it may not be necessary to look at derived words and derived entities, all the techniques we have used have taken care of the already. Here, we are going to look at them anyway.

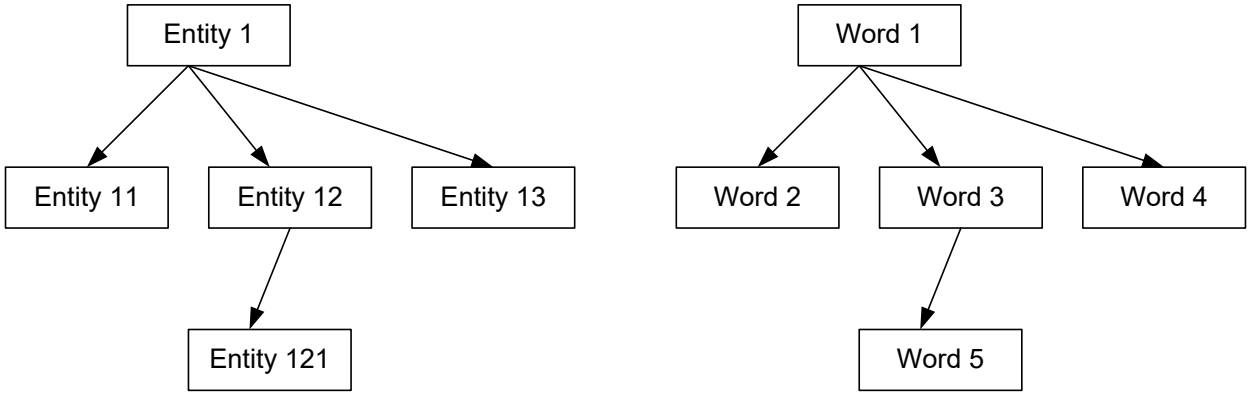
We already know that if an entity does not exist, then parts of that entity do not exist. In this case, any words that identify that entity and parts of that entity do not exist at all. As an example, let's take a look at the model below.



From the model above, *Entity 1* is viewed as the main entity and it does not exist. In this all parts of that entity do not exist as well. The parts of that entity can be viewed as entities derived from the main entity. Since words are used to identify entities, in this case, a word that is used to identify the main entity does not exist nor do words that identify parts of the main entity. The way to look at it, the words that are used to identify parts of the main entity are derived from the

 Copyright 2023 *The Speak Logic Project*

main entity. The words that are used to identify parts of entity are derived from the main words. We can call those words sub words. It is always good to represent them in the form below.



From the model above, we can see that *Entity 1* is viewed as the main entity, where *Word 1* is used to identify that entity. In this case, *Word 1* is viewed as the main word. If *Entity 1* does not exist, *Word 1* does not exist also because *Word 1* does not have an entity to identify. That also applies for all the sub entities and the sub words as well. For example, *Entity 11* is derived from an entity that does not exist, so *Word 2* cannot be used to identify that entity. There is not limit in that hierarchy. It goes indefinitely.

By understanding the explanation above, we can see that *if an entity does not exist, a word that is used to identify that entity does not exist as well. If entities do not exist, words that are used to identify them do not exist*.

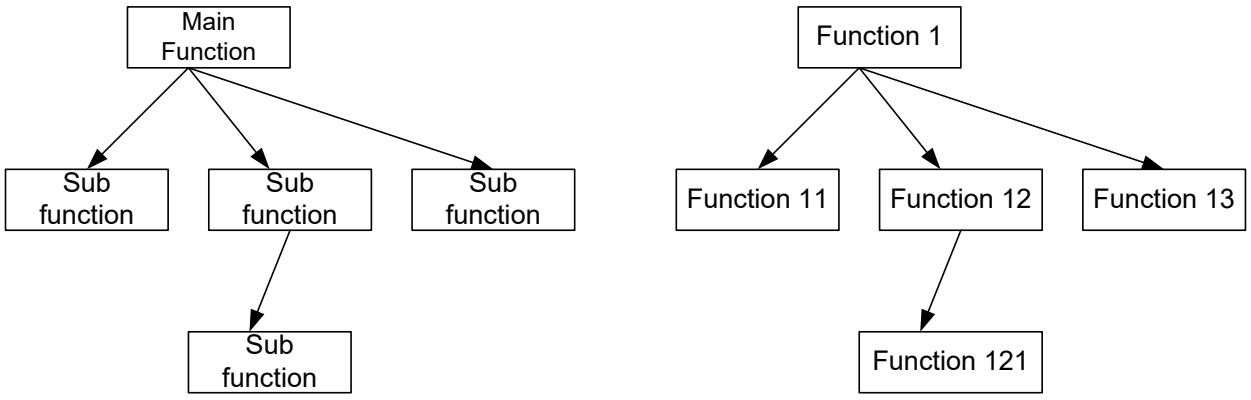
From the model above, we can see that the sub entities are derived from the main entity, where they are parts of the main entity. In this case, we can see that *if an entity does not exist, any entity derived from it does not exist at all.* In other words, entities that are derived from entities that do not exist, do not exist as well. Since words are used to identify entities, *if a word does not exist, any word derived from it does not exist at all.* This is the say as saying that, *words derived from words that do not exist, do not exist at all.*

To better understand the overall explanation, let's take it like this with a possible example. In terms of words, we can take it like this. A bad word or a negative word is viewed as a word that does not exist because it does not identify an entity or it improperly identifies it. In this case, any word derived from that word does not exist at all and also does not identify an entity and improperly identifies it. Another way to look at it, assume that a negative word or bad word is used, and it does not identify an entity or improperly identify it. From that word, a lot of sub words are derived, which we can call parts of that word. Since the main word is bad, parts of the main word which are viewed as sub words are bad as well. They do not exist and do not identify an entity, because the main entity does not exist.

 Copyright 2023 *The Speak Logic Project*

Related to visual aspect of communication, if the main entity is not visible, entities that are parts of that entity are not visible as well. In other words, if the main entity is not visible, entities that are derived from it are not visible as well. Let's take the model above as an example. If *Word 1* does not visually identify *Entity 1*, then all words derived from that word do not visually identify any entity derived from entity 1 or parts of *Entity 1*. For example, *Word 2* does not visually identify *Entity 11*, *Word 3* does not visually identify *Entity 12* and so forth. The way to look at it, *if a word does not visually identify an entity*, *any part of that word or sub words do not visually identify part of that entity*. This is the same as saying, *if a word does not visually identify an entity, any word derived from that word does not visually identify part of that entity or entity derived from that entity*. *If a main word does not visually identify a main entity, any word derived from that word does not identify any entity derived from that entity*.

Since functions are considered to be entity themselves, the overall technique applies for functions as well. For example, if a function does not exist, any function derived from it which is viewed as subfunction does not exist as well. To better understand this process, let's take a look at the model below.



From the model above, to the left all the functions below the main function are considered to be subfunctions or part of the main function. In this case, the existence of the main function enables the existence of the subfunctions. If the main function does not exist or it is negative, then the sub functions do not exist as well, and they are viewed as negative too.

From the model to the right, we identify the main function as *Function 1*, where the functions below are considered to be functions derived from the main function. For example, *Function 11* is derived from *Function 1,* and *Function 13* is derived from *Function 1* as well. In this case, if *Function 1* does not exist, *Function 11* which is derived from *Function 1* does not exist as well. If *Function 1* is not visibly identified, *Function 11* is not visibly identified as well. Since the main function does not exist, any word that we used to identify the main function does not exist

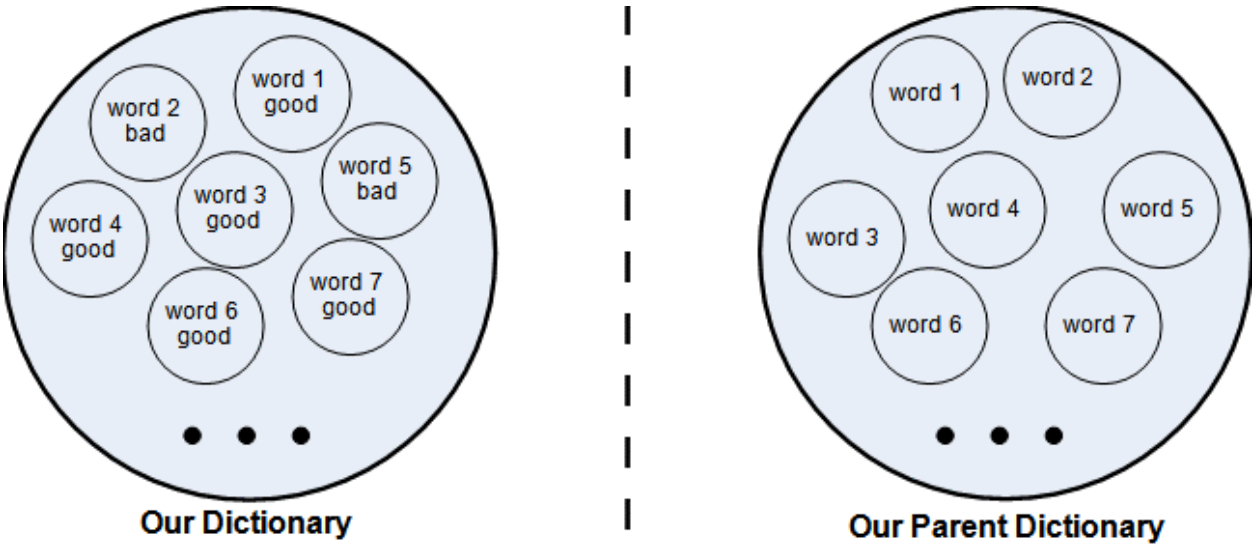
 Copyright 2023 *The Speak Logic Project*

as well. Any word that we used to identify the subfunctions, does not exist as well. Any words that we derived from the main word to identity the sub functions do not exist too because the main function does not exist.

Another way to look at it, the main function can be a negative function. What do we mean by a negative function? It is a function when it is executed it enables life to function abnormally. Since the main function is negative, any part of that function or subfunction is considered to be negative as well. Words that are used to identify that function and parts of that function are negative as well. In this case, the correction process enables changing that function to a positive function where parts of that functions are replaced by positive functions. In this case, the negative words that are used to identify that functions are replaced by positive words. The way to look at it, a function is negative, rather than adding more negative subfunctions to it or adding more functions to it, we simply change it to a positive function. This is the same as saying that, if the main word is negative rather than continuing to use it, we drop it and substitute it with another word.

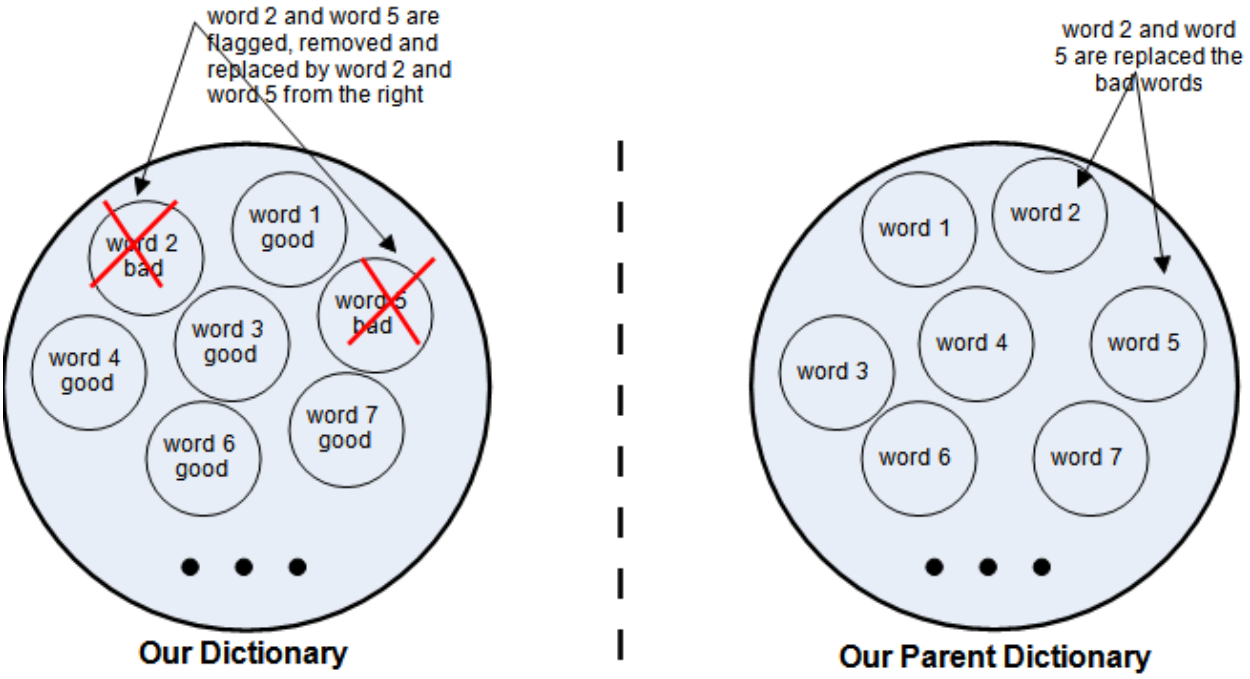
Conclusion

While we have shown and used many approaches to reduce the size of our dictionary to match the size of our parent dictionary, this is basically what we have been doing here. This is the problem we tried to solve. Verify or show that our dictionary is equal to our parent's dictionary. In other words, we need to show that our vocabulary is equal to all words in our parent's vocabulary. In this case, what we have done is we replaced all words in our dictionary to words from our parent's dictionary as shown below.



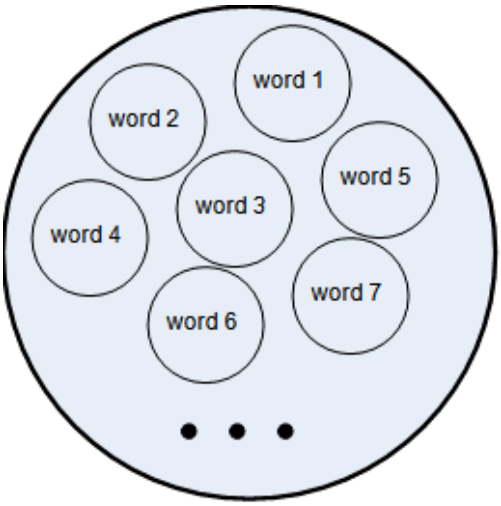
 Copyright 2023 *The Speak Logic Project*

From the model above, we show both dictionaries, where below we show the words replacement process. As we can see above, our parent's dictionary contains only good words and our dictionary contains both good words and bad words. In this case, we simply replace the bad words in our dictionary to good words from our parent's dictionary.

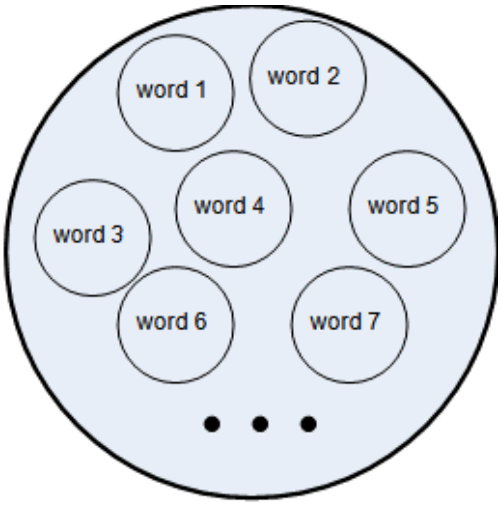


From above, by replacing the bad words from our dictionary to words in our parent's dictionary, we simply enable the size of our dictionary to be equal to the size of our parent dictionary. Another way to look at it, by understanding the error correction model and process, we can see that all good words in our dictionary belong to our parent's dictionary. In this case, the replacement process enables us the substitution of only words that do not belong to our parent's dictionary. The result of this process enables our dictionary to be equal to our parent's dictionary as shown below.

Copyright 2023 *The Speak Logic Project*



Our Dictionary



Our Parent Dictionary

This is basically what we have been doing here. By replacing all words in our dictionary to words from our parent's dictionary, we then reduce the size of our dictionary to be equal to the size of our parent's dictionary. Doing so enables our communication to be portable. It also enables us to be self-sufficient by performing our own corrections and enables us to communicate with our parent as well.

Some of you may be thinking, why no example of such words is given? The way to look at it, there is no need to. It is not important or necessary to do so. As we keep learning the principle and make progress in our communication, we will have a feeling of those words. They will come to us automatically. There is no need to have examples of those words. This is why no examples of such words are given. This will help us with the learning of the principle.

Why a Bigger Dictionary is not Needed

We have already explained the reason we do not need a bigger dictionary or vocabulary to communicate. We already know that we don't need a bigger dictionary because it does not help us solve any problem. Here, let's determine or verify why.

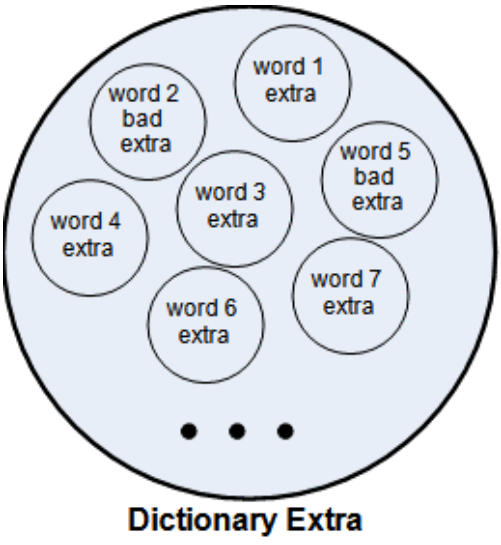
Consider the following example: if a bigger dictionary or vocabulary is needed, it must help us solve an identified problem. If we need a bigger dictionary or vocabulary to communicate, it must help us solve our problems. If our bigger dictionary does not help us any problem, then we don't need it at all. If we need a bigger dictionary to help us communicate, then we can identify problems that dictionary helps us solve. In this case, we can have a table similar to the one below where we can identify words in that dictionary and specific problem usage of those words help us solve.

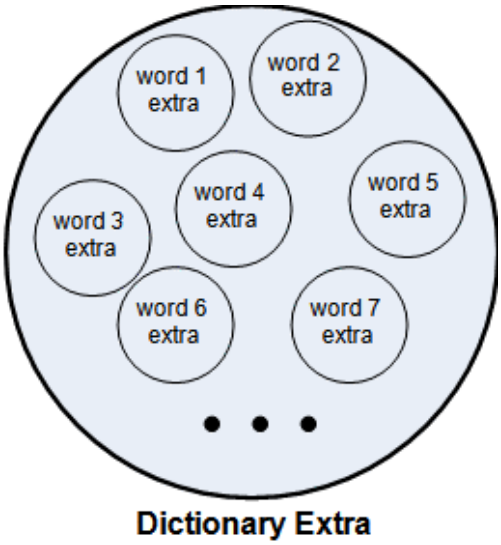
 Copyright 2023 *The Speak Logic Project*

Words	Problem Identified	Solve (Yes/No)
Word 1	*Problem 1*	No
Word 2 bad	*Problem 2*	No
Word 5 bad	*Problem 5*	No
Word 8	*Problem 8*	No
⋮	⋮	⋮

From the table above, if we can identify a word in our dictionary and that word does not help us solve any problem, then it is not needed. The words identified from the Words column above, are words from our original dictionary before reduced. Those words are considered groups of extra words that are not needed. In this case, if we can identify problems from the Problem Identified column that are solved by those words, then they are indeed needed in our dictionary. If we cannot identify problems solved by those words, then they are not needed as shown by the table above. This is the reason we do not need to increase the size of our dictionary because that does not help us solve any problem.

Another way to look at the example above, let's take it like this. We already know the size of our dictionary is bigger than the size of our dictionary. We already know that we want to make the size of our dictionary equal to the size of our parent's dictionary. In this case, our dictionary contains extra words that are not included in our parent's dictionary. Here, we need to determine whether those extra words help us solve identified problems. We can represent those extra words as a portion of our dictionary. In this case, we can simply call it *Dictionary Extra* as shown by the model below.

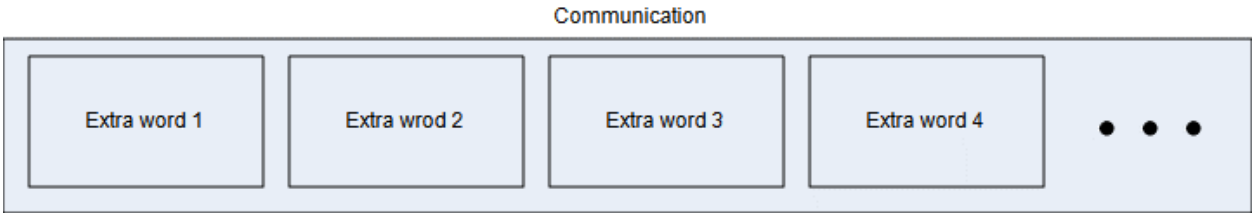




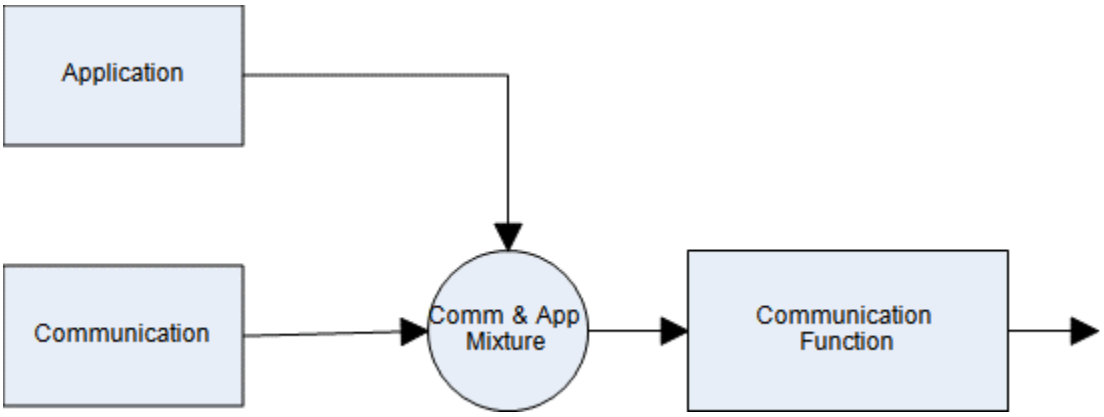
 Copyright 2023 *The Speak Logic Project*

From above, we simply represent the extra words from our dictionary as a dictionary by itself. We simply use number with the words to identify them and use extra with the words to denote that they are extra words from our regular dictionary. To the left, it shows that the extra dictionary also contains bad words. To the right, we identify either words as extra it does not matter if a word is bad.

From the table above, we have identified multiple problems that we need to solve. For example, we have *Problem 1*, *Problem 2*, *Problem 5*, *Problem 8*, and so forth that we need to solve. Now, if we can communicate by using those extra words to solve *Problem 1*, *Problem 2*, and so forth, then we can see that they are needed in our dictionary. In this case, we can identify that our communication is made of multiple extra words as shown below.

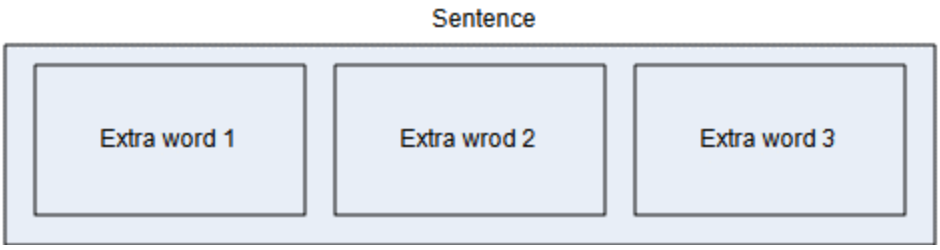


As shown by the model above, if the communication using those extra words enables us to solve problems, they are needed in our dictionary. To better understand that, let's take it like this. Our application, is given by the model below.

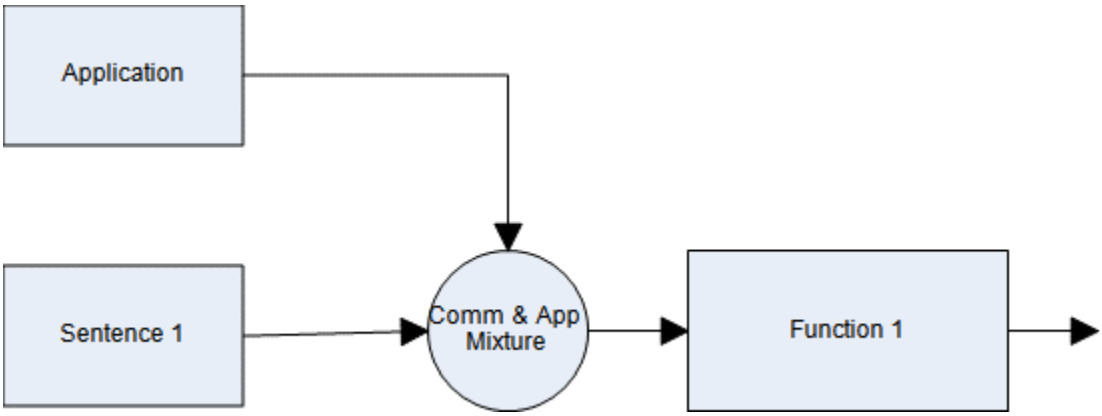


Now, let's assume that our communication is made of a single sentence that includes: *Extra Word 1*, *Extra Word 2*, and *Extra Word 3* as shown below.

 Copyright 2023 *The Speak Logic Project*



Using the application model above, we can simply replace the communication with that sentence. Let's name that sentence, *Sentence 1*. In this case, we can represent the model of our application with that sentence below.



From the model above, *Sentence 1* is viewed as the communication of our application, where *Function 1* is the function of our application. What is important here? In order to determine if those extra words are needed in our dictionary, we need to determine whether the execution of *Function 1* enables us to solve *Problem 1* identified from the table above, if so they are needed if not, they are not needed.

The way to look at it, if *Function 1* can be executed to solve *Problem 1*, then *Extra Word 1*, *Extra Word 2*, and *Extra Word 3* are needed in our dictionary; if not, they are not needed. The way to look at it, if we can communicate with words in our dictionary to solve identified problems, then those words are needed in our dictionary. However, if we cannot communicate with words in our dictionary to solve identified problems, then those words are not needed in our dictionary. From above, we can see that the extra words are not needed in our dictionary if they don't help us solve any problem. They are only needed if they can help us solve identified problems.

Another way to look at it, *if a problem exists and it is solved only by our parent dictionary, any additional dictionary is not needed*. To better understand that, if we identify a problem like *Problem 1*, where *Problem 1* is solved by a communication function such as *Function 1* where

 Copyright 2023 *The Speak Logic Project*

the communication is represented by a sentence with multiple words, where all the words in that sentence belong to our parent's dictionary, in this case, our parent's dictionary contains all the words needed to solved the identified problem or simply eliminate it when that function is executed.

What is important here? Any additional dictionary or dictionary that contains other words does not exist, since only our parent's dictionary enable us to solve the identified problem. In this case, we have similarities within the words that make up the sentence.

Reducing Dictionary Size Related to Time

We already know that a bigger dictionary does not help us or do us any good if we cannot solve problems with the extra words. We also know that having a big dictionary does not help us at all, if we still need our parent to give us feedback to correct our sentences. In addition to that, we have also learned that having a bigger dictionary is not event important for us if we cannot communicate with our parent.

By understanding the paragraph above, we can see that since the extra words that increase the size of our dictionary do not help us solve problems, correct our sentences, and enable us to communicate with our parent, we need to eliminate the extra words. In other words, the extra size that makes our dictionary need to be reduced to zero, since it serves us no purpose.

Since a principle cannot be learned instantly, it is not possible for us to reduce the number of extra words in our dictionary to zero as we speak. It is not possible or practical. In other words, the process of learning the principle of communication is not instantaneous, so it is not possible or practical to reduce the size of our dictionary to the level of our parent. However, as we keep learning the principle and make progress in our understanding, we can reduce those extra words little by little related to time. To better understand this process, let's take a look at the model below. The model below is viewed as the process of removal.

Time				
Time 1 5% Of the unneeded words are removed	*Time 2* 10% Of the unneeded words are removed	*Time 3* 15% Of the unneeded words are removed	*Time 4* 20% Of the unneeded words are removed	• • •

 Copyright 2023 *The Speak Logic Project*

From the table above, we need to remove a percent of those unwanted words related to time. In this case, we cannot remove all of them instantly, but as time goes, we can remove some of the words which reduce the size of the unwanted words.

For example, let's assume that we have a total of 1000 unwanted words to remove in our dictionary. In this case, we cannot remove all of them instantly. We start at *Time 1*, where we remove 5 % %of them. Then we have 950 extra words left to be removed. At *Time 2*, we remove another 10% of the 950, then we have 850 remaining to be removed. If we continue that process, at some point in time, we expect to remove all of them. The table below provides more explanation about this process.

Time				
Time 1 We remove 5% of the 1000 extra words so we have 950 remaining to be removed	*Time 2* We remove 10% of the 950 remaining extra words, so we have 850 words remaining to be removed	*Time 3* 15% If we remove 15% of the remaining, we have less remaining words to be removed	*Time 4* 20% If we remove 20% of the remaining, we have much less left to be removed	• • •

From the model above, as we start learning the principle and make progress in our learning, we remove 5% of the list of unneeded words. In this case, the list is reduced to a lower level. As we continue learning the principle and make progress in our learning, at a letter time, we remove more words from the unneeded list, which further reduce the size of unneeded words. As we can see, we start with 1000 words, then we remove 5% of that which is 50 with a remaining 950 from the unneeded list. Then as we make progress in our learning, we reduce an addition 10% in the unneeded list, which removes 95 unneeded words from the list of 950 unneeded words. If we continue learning the principle, at a later time we should remove all 1000 unneeded words completely.

Let's take a look at the example below which is the continuation of the one above. For the example below, let's assume that we have 1000 words to remove related to time.

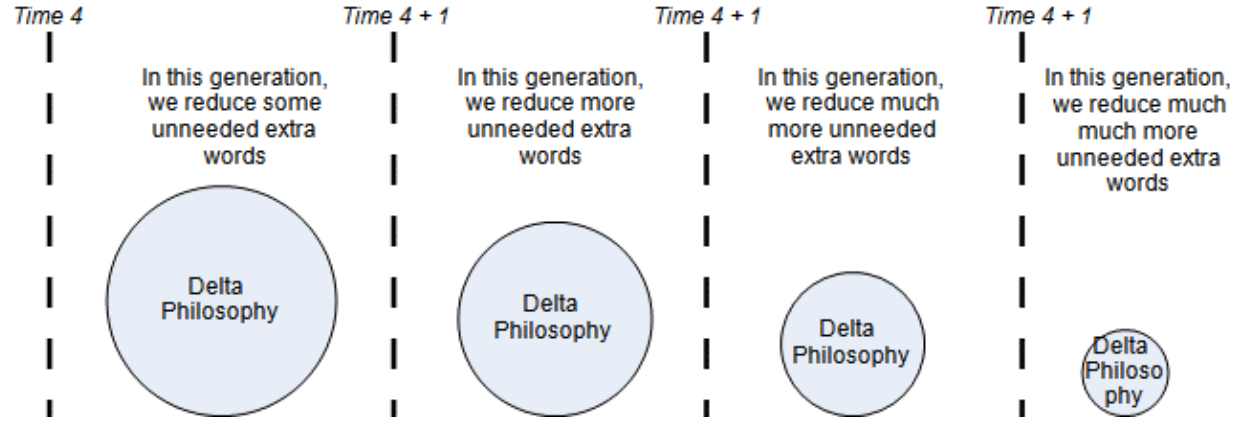
Time

Copyright 2023 *The Speak Logic Project*

Time 1	*Time 2*	*Time 3*	
We remove 10 of the 1000 then we have remaining to remove	We remove 15 of the 990 then we have remaining to remove	We remove 20 of the 975 then we have remaining to remove	• • •
We remove *Word 1* to *Word 10*, so we have *Word 11* to *Word 1000* left to be removed	We remove *Word 11* to *Word 25*, so we have *Word 26* to *Word 1000* left to be removed	We remove *Word 26* to *Word 45*, so we have *Word 46* to *Word 1000* left to be removed	

From above, at *Time 1* we remove specific number of words from the list of unneeded words and we left from less to be removed from where we start. As we can see, we started with 1000 to be remove, then we take it down to 990. At *Time 2*, we removed 15 of the 990, and we take it down to 975. As we can see, if we continue doing that related to time, at some point in time, we expect to remove all those unneeded words from our dictionary. At that time, our dictionary will be equal to our parent's dictionary.

The number of words in our dictionary is related to philosophy inheritance. The more philosophies we inherit, the bigger is the size of our dictionary. The reduction of philosophy enables us to reduce the size of our dictionary. The chart below shows an example on how the process works.



As we can see from the model, the reduction of words in our dictionary happens in a generation basis. As we make progress learning the principle, philosophies are reduced so does the number

 Copyright 2023 *The Speak Logic Project*

of unneeded words in our dictionary. In this case, we can see as philosophies come down, the number of unneeded words in our dictionary comes down as well as the size of our dictionary.

At some point in time, as we replace all philosophies with our parent principle, all the extra words are removed from our dictionary and at that time, our dictionary will be equal to our parent's dictionary. As shown from the model above, the reduction happens on a generation per generation basis.

Copyright 2023 *The Speak Logic Project*

Understanding Complexity and our Vocabulary

Introduction

From the previous chapter, we have learned that multiple techniques enable us to reduce the size of our dictionary to be equal to the size of our parent dictionary. We have also learned that the increase of the size of our dictionary does not do us any good since it does not enable us to solve problems. The increase of the size of our dictionary simply enables our dictionary to be too big and less compact. By reducing the size of our dictionary, that enables it to be more compact and portable.

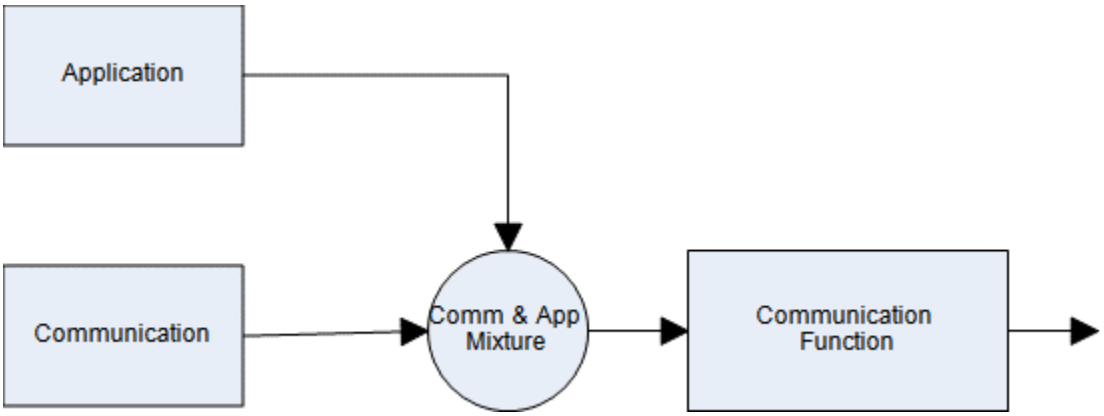
If we have a big dictionary and our parent still corrects our communication, what good does it do for us? If we have a big dictionary and our sentences are still unportable what good does it do for us? If we have a big vocabulary and we cannot take it anywhere we go, what good does it do for us? If we have a big vocabulary and we still cannot communicate with our parent, what good does it do for us? If we have a big dictionary and it does not help us solve our problems, what good does it do for us? In this case, we can see that a big dictionary or vocabulary does not do us any good; it simply enables the increase of complexity.

In this chapter, we are going to look at complexity in relationship to the size of our dictionary. We are going to look at how the increase of the size of our dictionary simply enables us to increase complexity in what we do. We are going to look at how the reduction of the size of our dictionary to be equal to the size of our parent's dictionary enables us to reduce complexity in what we do.

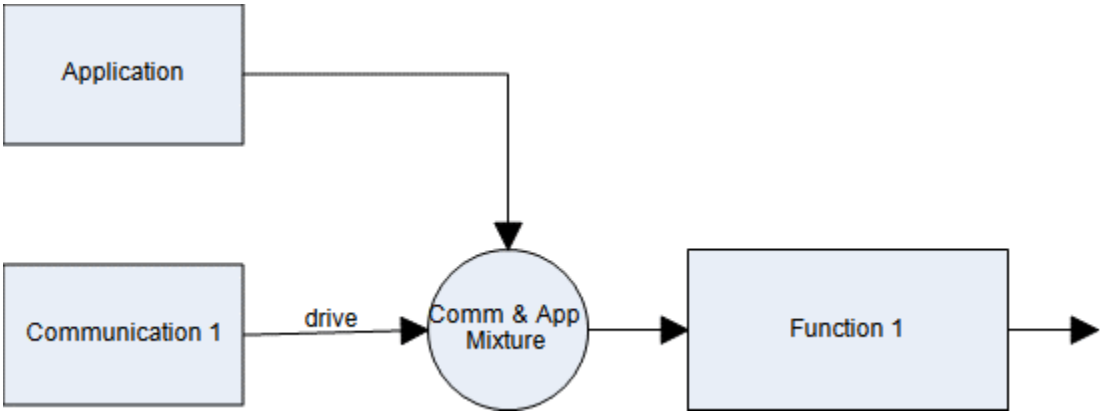
Copyright 2023 *The Speak Logic Project*

Complexity in Size of our Dictionary

The communication function which is the function of our application was given to us by the model below.



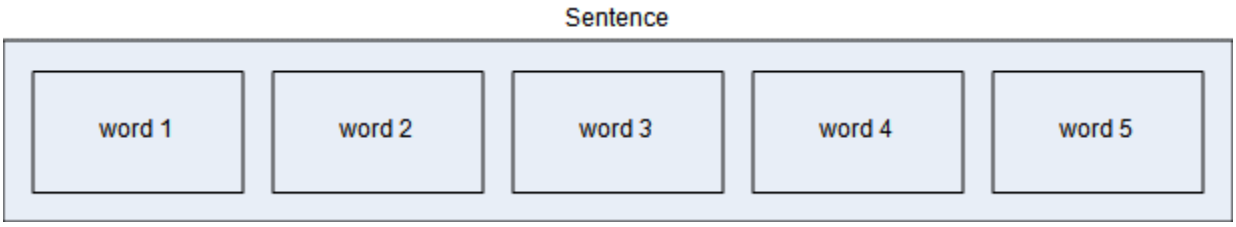
where the communication to the left is the communication of people in the application. In this case we can say that the communication function to the right is naturally driven by the communication to the left and we can show that in the model below.



From the model above, *Communication 1* is viewed as the natural communication that enables *Function 1* to execute; where *Communication 1* is the communication of people in the application. What do we mean by natural communication for the application? We mean *Communication 1* is the only communication needed by *Function 1* to execute the application.

For a typical application, let *Communication 1*, the communication of the application, be a complete sentence. In this case we have:

 Copyright 2023 *The Speak Logic Project*

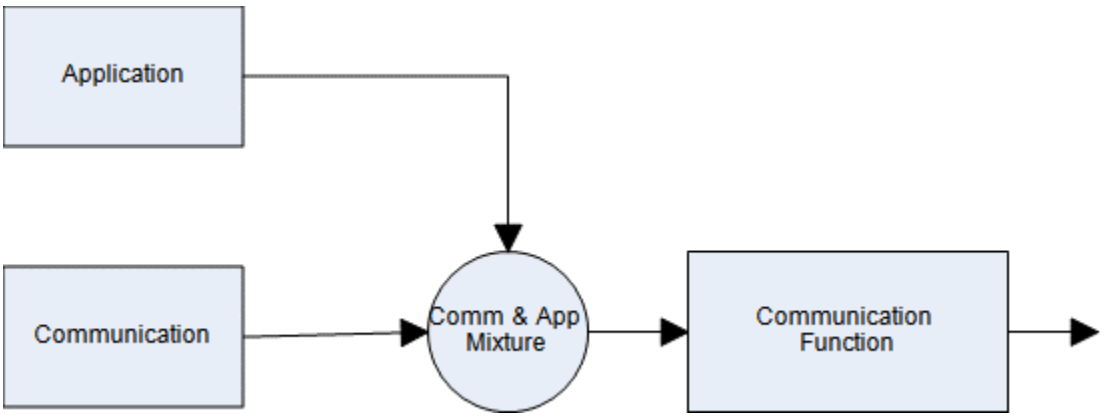


where *word 1* to *word 5* are words the sentence is made up. What is important here is that *word 1* to *word 5* are considered to be words that drive that application. We can say that those words are natural to the application. In this case, the application is naturally driven by *word 1* to *word 5*.

From the previous chapter, we have learned that if the increase of the size of our dictionary does not help us solve any problem, then it serves us no purpose. From that and in relationship with the error correction process, we can see that only our parent's dictionary enables us to solve identified problems. Since our goal is to equalize the size of our dictionary to the size of our parent's dictionary, in this case the size of our parent's dictionary enables us to solve identified problems. By understanding that, we can see the good words used in our dictionary or vocabulary belong to our parent's dictionary. By understanding that, we can see that *word 1* to *word 5* belongs to our parent's dictionary.

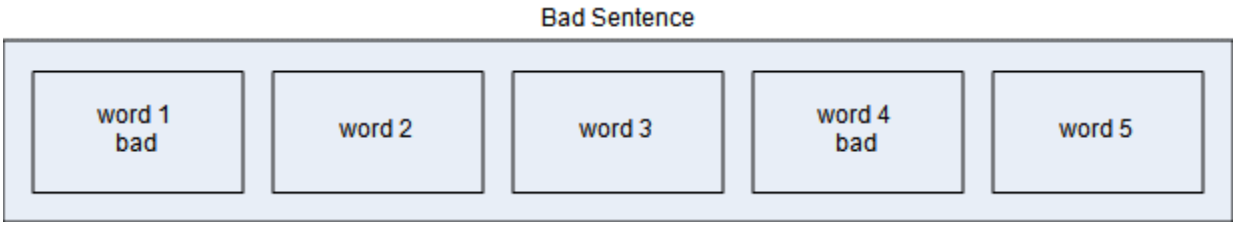
By understanding the explanation above, we can see that our parent's dictionary is viewed as our natural dictionary. In this case we can say that the words that we used to naturally drive our application belong to that dictionary. In other words, our application is naturally driven by its natural communication where that communication is made of words from the dictionary given to us by our parent. From that, we can also see *if our application is driven by words from our parent's dictionary, then it is naturally driven*.

In terms of complexity, we can say that the function of our application, which is the communication function represented by the model below:

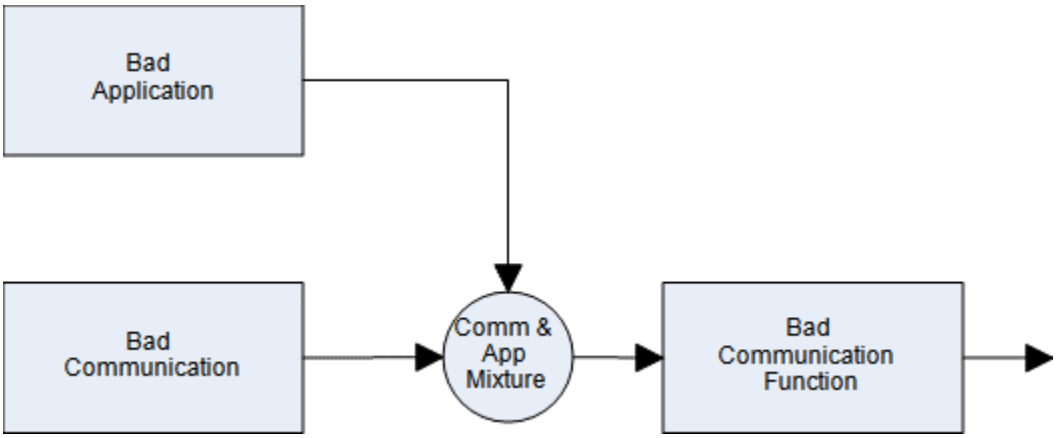


 Copyright 2023 *The Speak Logic Project*

If only the application is naturally driven by the communication which is viewed as collection of words in our parent's dictionary. If the application is not driven by its natural communication, then we have for example:



What is important here is that the communication above is no longer natural to that application. In this case, the function of our application becomes:

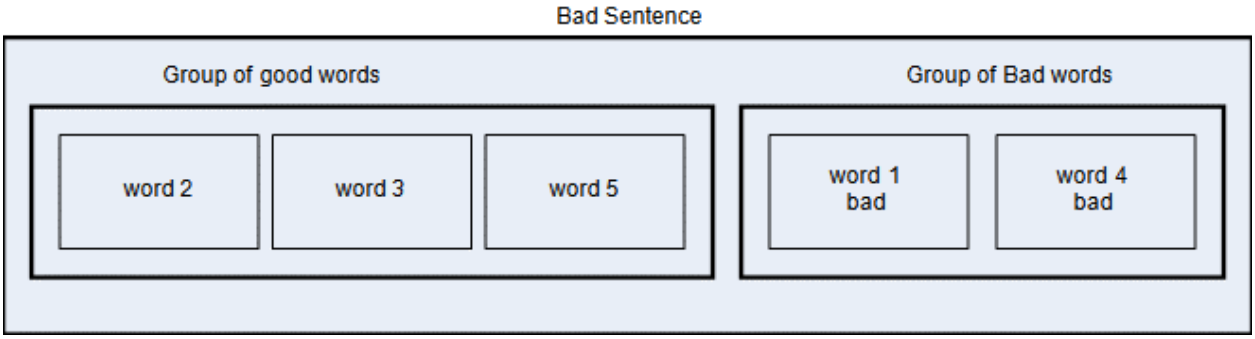


From the list of words that make up the sentence above, we can see that all the words that make up that sentence do not belong to our parent's dictionary. In this case, we have the following: *word 1* does not belong to our parent's dictionary, *word 2* belongs to our parent's dictionary, *word 3* belongs to our parent's dictionary, *word 4* does not belong to our parent's dictionary, and *word 5* belongs to our parent's dictionary.

In terms of complexity, we have to look at the error correction function. To execute the application, while we communicate about executing our application, our parent heard what we say where what we say is incorrect and provide us with feedback to make the correction. What is important here is that in terms of the Error Correction Function (ECF), before we have *Bad Communication Function* and after we have good *Communication Function*. In this case, it looks like we were working on doing something, then we were in the wrong direction, then we received feedback to do it properly then we had to make adjustments to what we were doing. In this case, we can see clearly that takes more effort from us which is viewed as complexity.

As we can see, when the application is not driven by its natural communication, that simply increases complexity in the application. To reduce complexity in the application, we need to

allow the application to be driven by its natural communication. In this case, we can make adjustments in our communication to enable the application to be driven by its natural communication. To do so, from the sentence above, we can identify both bad words and good words into two groups. From that, we can partition the sentence in this form:

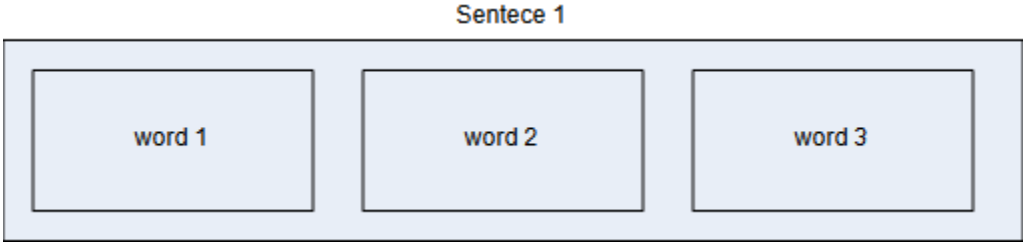


where the group of good words naturally exists in our parent dictionary and the group of bad words are not. Using the Error Correction Function enables us to make adjustments to the sentence so it can be in its natural form to enable the application. Doing so enables the reduction of complexity in our communication which enables us to reduce complexity in the application.

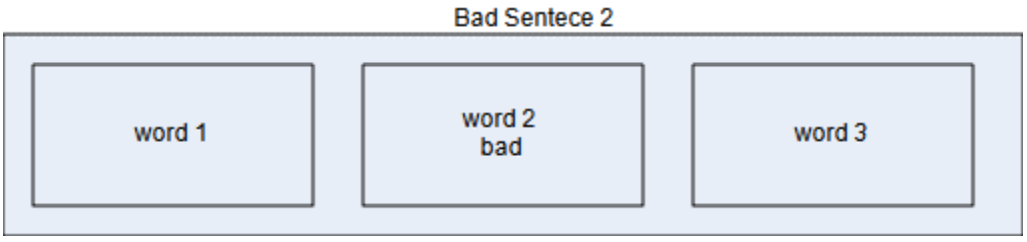
To better understand how the size of our dictionary increases complexity in our application, we can think it as complexity of our communication. By understanding the previous chapter, we can see that the complexity in our communication is related to the size of our dictionary. In this case, reducing complexity in our communication simply decreases the size of our dictionary. To better understand that, let's take it like this. From our parent, we have learned that, *if there is a relationship between all words in a sentence then that sentence is correct*. In terms of complexity, it is easier for us to look at a relationship between all words in a sentence than multiple relationships.

All of the examples that were given to us from the previous chapter enable us to reduce complexity in our communication so complexity can be reduced from our application. By reducing complexity in our communication, we simply put our communication into its natural form so it can drive the application to solve the underlying problem.

As another example, in terms of words in our dictionary let's assume that we have two sentences:



 Copyright 2023 *The Speak Logic Project*



Now, let's show that *Sentence 2* creates complexity. If the number of words in our dictionary is 100 and the number of words in our parent dictionary is 50. In terms of communication, as we increase the number of words in our dictionary, we can see complexity increase. As we reduce the number of words, in our dictionary we can see complexity decrease. For example, if we remove all negative words, then we have less words to think about in our communication which reduce complexity. For example, using a smaller size dictionary or vocabulary to communicate is less complex than using a bigger size dictionary. The bigger size takes more efforts from us than the smaller size. Since the smaller size enable us to execute our application efficiently, we can see that the smaller size is natural to our application which enables us to reduce complexity in our application.

Summary

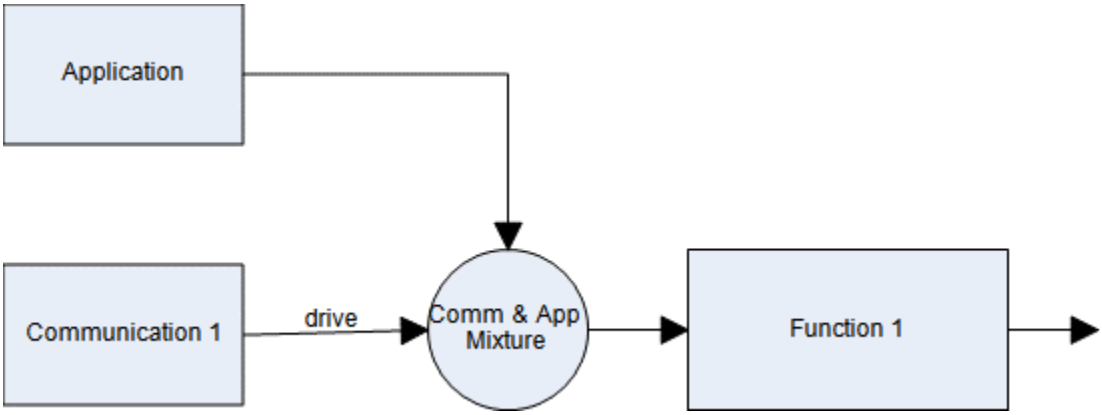
Since our parent dictionary includes less words, and our parent uses it to provide us with feedback to correct our communication, we can call our parent's dictionary a natural dictionary or a natural form of dictionary. The way to look at it, naturally our application is driven by our communication, where the communication that enables it is a natural communication for that application. If a sentence includes all words from that dictionary, we can see that sentence is the natural sentence that drives the application.

Our dictionary contains much more words than the number of words in our parent's dictionary, where our parent's dictionary is natural and our dictionary is not. In this case, our dictionary simply increases complexity in our application. By reducing the size of our dictionary to the size of our parent's dictionary, we simply reduce our dictionary to its natural form which enables our application. By doing so, we simply reduce it to the same form of our parent's which is the natural form. When we reduce the size of our dictionary to match our parent's dictionary, we simply reduce our dictionary to natural form which enables us to reduce complexity. The following steps show how increase the size of our dictionary simply increases complexity.

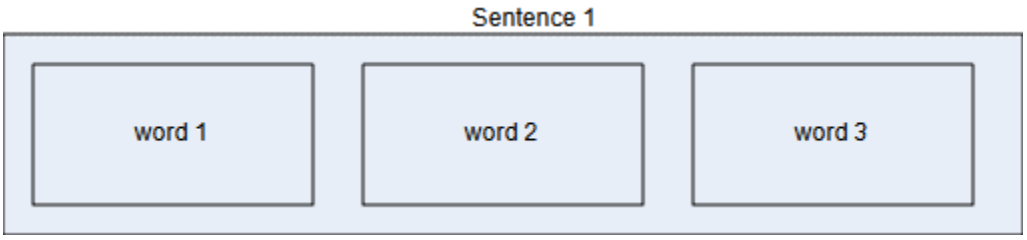
- Increasing number of words in our dictionary increases complexity.
- Increasing number of words in our dictionary to form sentences simply increases complexity.

 Copyright 2023 *The Speak Logic Project*

- When we increase number of words in our dictionary or vocabulary, natural communication does not drive our application. In this case, our application is not driven by its natural communication.
- Increasing the size of our dictionary simply adds more burden to us where our application is not successful upon.
- The increase of words in our dictionary enable us to increase complexity which allow us to commit more effort with little or no success.
- Since the introduction of additional communication in our application enables our application to become more complex, by reducing the number of words in our dictionary, that enables us to include less communication in our application which enables us to reduce complexity in our application. In other words, if we reduce the number of words in our dictionary, we would add less unwanted communication to our project or application which enables us to reduce complexity.
- To better understand the part above, for our application we have:

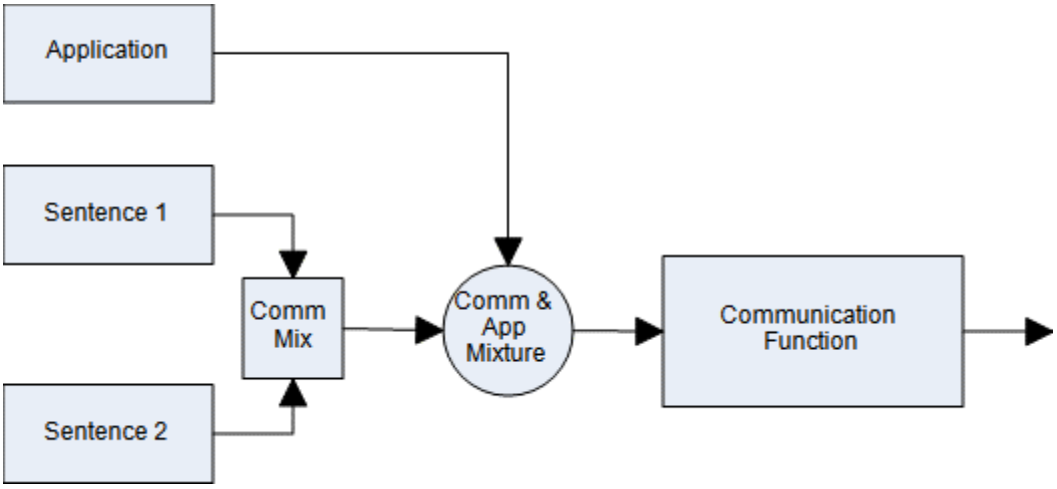


where *Communication 1* is viewed as a sentence in the form below, where *Communication 1* is identified as *Sentence 1*.

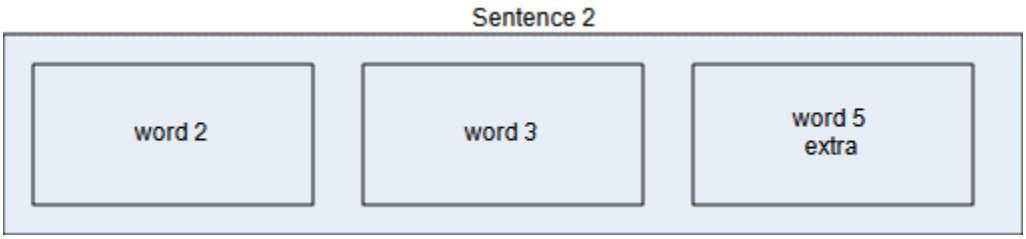


Sentence 1 above is related to the project. If additional communication is added to the project as shown by the model below:

 Copyright 2023 *The Speak Logic Project*



From the model above, the second added communication is viewed as *Sentence 2*, where it is made of the following words.



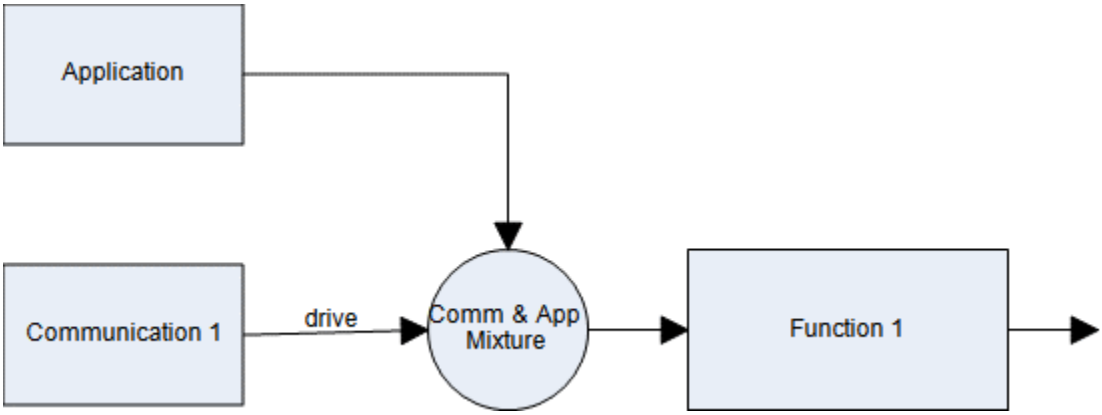
where *word 5 extra* is extra word added from the dictionary which makes the second sentence above becomes unwanted for the application. By reducing the size of the dictionary to remove the extra added word, we simply reduce the size of our dictionary which enables us to remove unwanted communication in our application. This process enables us to reduce complexity in our application. The extra words in the dictionary enables extra communication to be added to the application that is not native or natural to the application which increase complexity in the application. Removing the extra words enables reducing unwanted communication which enables reduction of complexity.

From the two sentences above, in the set of words used for the second sentence, if *word 5 extra* is removed from the dictionary or vocabulary or replaced by another word, that further reduces the size of the dictionary. In this case, the extra word simply increases complexity in the communication which increases complexity in the application. By removing complexity or the extra word, we have a second sentence that is similar to the first one. What is important here is that looking for similarities in the communication enables us to reduce complexity in the application. Identify similarities within the

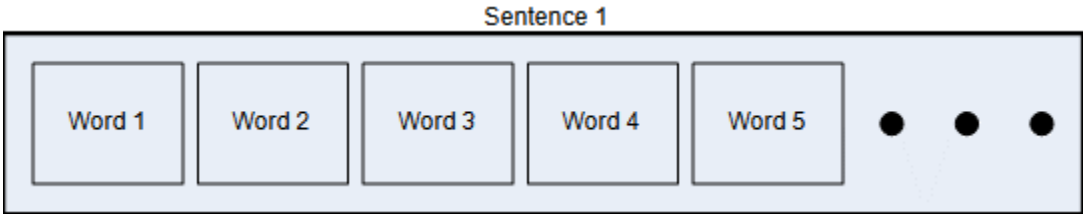
 Copyright 2023 *The Speak Logic Project*

communication enables the communication to be similar which reduces complexity in the application.

- If a problem exists and it is solved only by our parent's dictionary, another dictionary is not needed. Any additional dictionary only enables us to increase complexity in our application or what we do.
- In terms of visual aspect of communication, only resources that we identify that are allocated for our application enables our application. Resources that are not allocated to our application do not enable our application. The communication about resources allocated to our application is natural to our application where the communication about unallocated resources is not natural to our application. When we communicate about unallocated resources to execute our application, we commit error in communication which increases complexity. To better understand that, let's take it like this:



where *Communication 1* is the communication for the application. It is natural to the application and related to the allocated resources. If we identify *Communication 1* as a *Sentence 1*, where



is related to the allocated resources for the application and it is natural to the application. In this case, all words that make up that sentence belong to our parent dictionary and all other communication that do not enable the application or about unallocated resources are

 Copyright 2023 *The Speak Logic Project*

not natural to the application. Those set of communication simply enable us to increase complexity. The way to look at it, communication about allocated resources enables us to match the size of our dictionary to our parent's dictionary or reduces the size of our dictionary to match our parent's dictionary where communication about unallocated resources enables us to increase the size of our dictionary.

Copyright 2023 *The Speak Logic Project*

Understanding Stability and our Vocabulary

Introduction

Previously, we have learned that by reducing the size of our dictionary to be equal to the size of our parent dictionary, that enables us to solve problems and communicate properly. We have also learned that by reducing the size of our dictionary to be equal to the size of our parent dictionary, enables us not to depend on our parent to correct our communication. When the size of our dictionary is equal to the size of our parent dictionary, that enables us to be self-sufficient and our communication would not need correction at all.

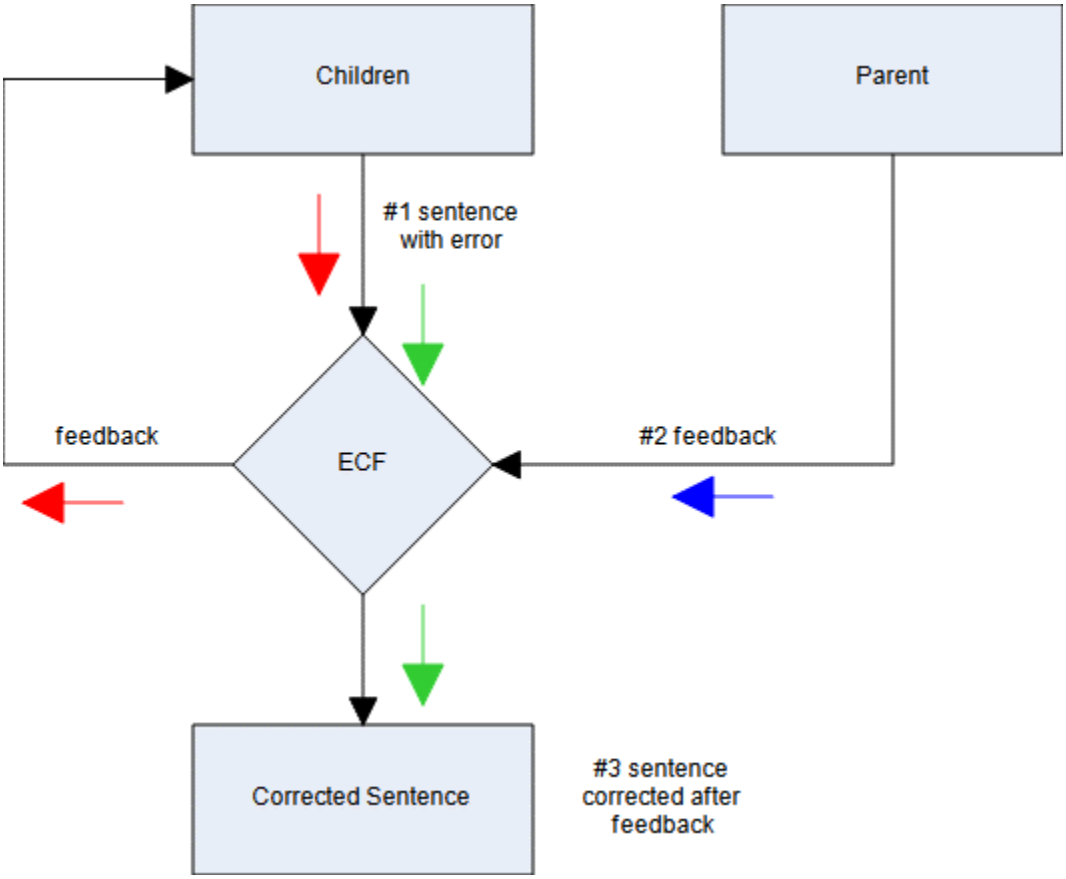
By looking at the model of the error correction process, we can see that our parent corrects us because of the errors that are included in our communication. In terms of stability, we can see that since our parent corrects our communication while we are talking, it looks like we are not stable enough to communicate with our parent or in front of our parent or where our parent can hear us or is aware of our communication. The way to look at it, if we feel confident enough to communicate without our communication getting corrected by our parent, then we feel that our communication needs no correction and we feel very stable because of that.

In this chapter, we are going to look at how the size of our vocabulary affects our stability. We have already learned that increasing the size of our vocabulary also increases complexity in our application and when we reduce the size of our vocabulary, we also reduce complexity in our application. In this chapter, we are going to learn how the increase of the size of our dictionary enables us to become unstable while equalizing the size of our dictionary to our parent's dictionary enables us to be stable.

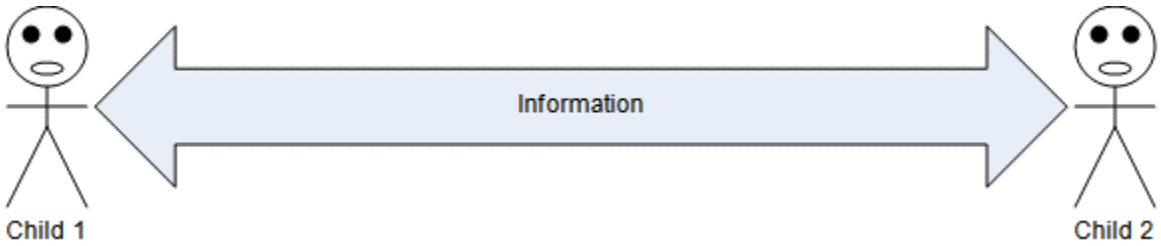
Copyright 2023 *The Speak Logic Project*

Stability and Size of our Dictionary

To better understand stability and the size of our dictionary, let's take another look at the error correction process as shown by the model below.



From the model above, let's assume two children: *Child 1* and *Child 2*. While both of them communicate together, one or both commit an error which get corrected by the parent. We can represent the communication of the children by the model below where they pass information to each other.

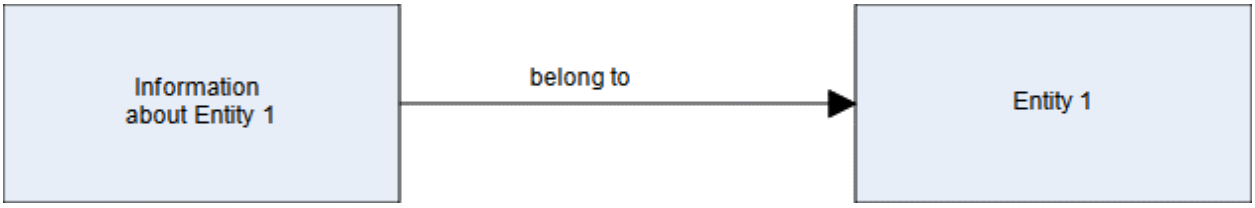


From the model above, *Child1* and *Child 2* can be replaced as the children of the feedback model above. As we can see, as the children are passing information to each other. If there is error in

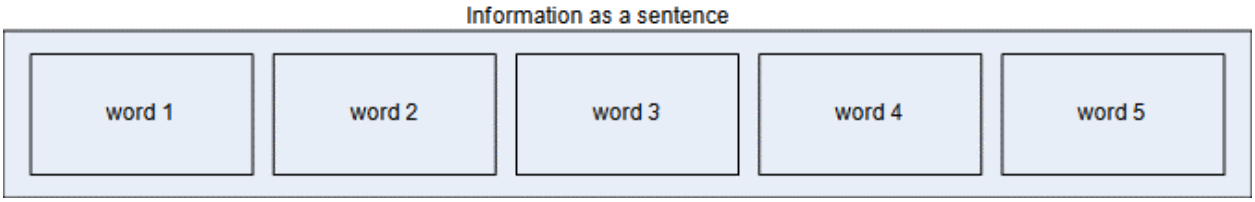
 Copyright 2023 *The Speak Logic Project*

the communication, they receive feedback from the parent to enable the correction. What is important here? In terms of information, the information about an entity enables the children to communicate and pass information to each other. During the communication process, that information does not change and it belongs to that entity. In this case, if the information that entity is about does not exist or that entity the information belongs to does not exist, the communication is not enabled.

By understanding the paragraph above, we can see that since the communication about the entity enables the communication of the children, then that information is natural to that entity and cannot be changed or adjusted. It is always better to represent that information in relationship about that entity by the model below.

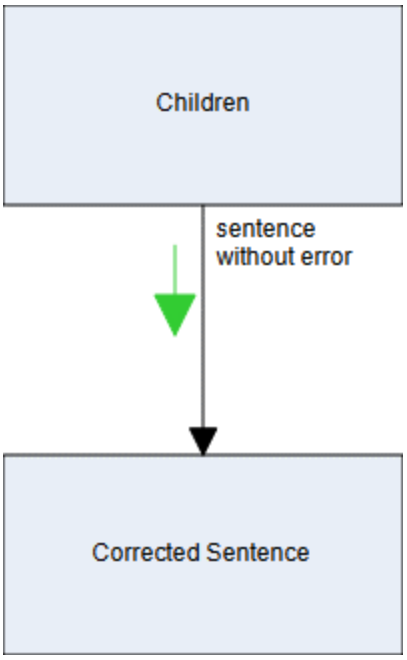


If we identify that information as a single sentence, then we can represent that information by the model below, where it is made of multiple words. From the model below, all the words are natural to that information and all of them belong to our parent dictionary. No additional words that are not part of that information is needed to present that information.



By looking at the two models above, we know that during the communication process, if the parent hears the children and no feedback is given, then the communication is correct. What is important here is that the feedback model above only exists to enable a correction if there is error in the communication. If there is no error in the communication, feedback is not needed, and the model does not exist. The way to look at it, during the communication, if our parent hears us and there is no feedback or alert, then the sentence is correct. In this case, all words used in that sentence belongs to our parent's dictionary. To better understand the nonexistence of the feedback model without error, let's represent it below with the absence of error.

Copyright 2023 *The Speak Logic Project*



As we can see from the feedback model, without error in the commutation, the Error Correction Function does not exist, so our parent does not appear in the model. In other words, when we are talking, if we don't commit error in our communication, our communication is correct and our parent does not alert us. During our communication, if our sentence is correct, it is acceptable by our parent and we have no problem with that. Another way to say it, if our communication is correct, it is acceptable by our parent and both us and our parent have no problem with that. As we can see, with a correct communication or a communication that needs no correction, both the children and the parent feel comfortable with that. As we can see, with a corrected communication, the children feel more stable. Another way to look at it, if we feel that we are not committing error in our communication, then we feel more stable talking while our parent is listen to us. In this case, we also feel more stable talking with our parent.

By understanding the explanation above related to the size of our dictionary, since when our communication is correct all words in that communication belong to our parent's dictionary, in this case if we can communicate only by using words from our parent's dictionary, then we feel more stable. If we can communicate where we don't use words to enable correction or receive feedback from our parent, then we feel more stable about our communication. As we can see, since the increase of the size of our dictionary enables us to receive feedback from our parent to get our communication corrected, we feel less stable with a bigger dictionary than with a smaller dictionary that is equal to our parent's dictionary. In this case, if we can reduce the size of our dictionary to be equal to the size of our parent's dictionary, we will not need to be corrected or receive feedback and we will feel very stable.

 Copyright 2023 *The Speak Logic Project*

Another way to look at stability and the size of our dictionary, is that from the vocabulary reduction chapter, we have learned that if a word in our dictionary does not promote stability, then we can remove it from our dictionary and replace it with a word that promotes stability. From that, we can see words that do not promote stability should not be included in our dictionary or vocabulary. In this case, removing those words in our dictionary enable us to become more stable. Let's say it again. In terms of stability, we have learned that if a word does not promote stability, it is not in our parent's dictionary and we have to remove it from our dictionary in order for the size of our dictionary to equal to size of our parent's dictionary. By equalizing the size of our dictionary to the size of our parent's dictionary, we simply promote stability or being stable. With that, we can see the communication of our parent always promotes stability as well as the words used by our parent.

As we can see from the feedback model, the fact that we depend on our parent to correct our communication, we feel very unstable. The way to look at it, the relationship between the two model above is that the feedback model shows that we are not stable by ourselves, while the second model shows that we are stable by ourselves.

Reducing the Size of our Dictionary and Stability

Previously we learned the effect of philosophy inheritance on the size of our dictionary. We learned that as inherited more philosophies related to time, the size of our dictionary increases tremendously. As we keep learning the principle, related to time the reduction of those philosophies enables us to reduce the size of our dictionary until the size of our dictionary is equal to the size of our parent dictionary. What is important here? Since philosophy inheritance enables the size of our dictionary to increase—philosophies go up, size of our dictionary goes up—and it enables us to be less stable, the increase of the size of our dictionary and philosophy inheritance enables us to become unstable as well. Therefore, as the size of our dictionary increases, our stability decreases.

Conclusion

All we have learned from this chapter is that the size of our dictionary affects our stability. As we reduce the size of our dictionary, we become more stable. As we reduce the size of our dictionary to equal to the size of our parent's dictionary, we become very stable at a point where we do not need our parent to correct our communication or provide us with feedback. At that time, we are stable enough to be in our own without depending on our parent.

 Copyright 2023 *The Speak Logic Project*

Index

%

% removal, 140

1

1000 unneeded words, 140
1000 words to remove, 140

3

3 entities of solving problems, 54

A

a person cannot communicate for another person, 57
absence of errors in communication, 42
acceptable communication, 156
action, 65
Action, 8
action taken, 53
Action Taken, 8
actual principle, 29
actual solution, 119
add people from other area to project, 62
adding more words, 103
adding negative to information, 101
additional dictionary, 138
adds sense to sentence, 48
adjustment communication reduce complexity, 147
alert, 155
all employees' functions, 16
all entities usage, 27
all people's functions, 16
allocated resources, 91
allow a person from another organization to execute functions, 58
allow other people outside organization, 58
allow other people to execute functions in our application, 59
allow other people to execute their functions, 59
ambiguity, 116
analysis enables understanding communication., 30
Analysis of Communication, 31
analyze problem statement, 65
analyze word, 51
analyze words in sentence, 52
analyzed by people in application, 65
appear in model, 59
Application 2 in Area 2, 65
application becomes less complex, 74
application correctness, 64
application driven by communication, 88
application driven by words, 145
Application Execution Location, 49
application function verification, 64
application idea and complexity, 74
application natural, 151
application naturally driven, 145
application needs other application, 60
application performance, 67, 81
application require another function, 58
application verification, 64
Application Verification, 63
application verified with problem statement, 65
apply principle, 51
apply principles to correct error, 50
apply the error correction function, 51
applying feedback, 96
Applying Principle in Application, 27
aspect of entity enables existence of entity, 113
aspect of principle to verify principle, 35
aspect of words, 113
assume functions, 57
at least a good word, 104
at least one good word, 52

B

bad context, 43, 47
bad context ad error, 43

 Copyright 2023 *The Speak Logic Project*

bad sentence, 47
bad words, 43, 47, 51, 94, 114
bad words as error, 43
bad words improper entity, 131
basis of correction, 52
become more stable, 157
better understanding of the principle, 59
big dictionary, 102
burden and complexity, 149

C

cannot assume or take other people's functions, 59
certain words, 94
changing the context of the sentence, 44
claim to solve problem, 128
collection of sentences, 111
collection of word, 111
collection of words, 45, 95
combination of words existence, 125
combination words identification, 126
combined words, 115
combined words to eliminate other words, 105
combining relationships, 37
commination comes after entity identification, 102
commination exists after entity identification, 102
common communication, 82
common communication reduce complexity, 83
common idea, 82
communicate about resources, 88
communicate with our parent, 153
communication about resources, 87, 90
communication application depends on, 70
communication common to each other, 83
communication contains error, 50
communication does not need correction, 156
communication domain, 56
communication drives application, 67
communication enabled, 112
communication enables application, 68
communication enables function, 144
communication error results to complexity, 72
communication errors about resources, 92
communication exists second, 100
communication flow, 11
communication function, 68
communication holder, 22, 24
Communication in the Application, 10
Communication Interactions, 17
communication of Ms. Smith, 59
communication of people at time, 13
communication of Peter, 59
communication principle as principle of operation, 34
Communication Principles, 32
communication result, 24
communication with errors disable application, 44
communication without errors enables application, 44
comparison of entities, 124
complete correction, 48
complex entity, 67
Complexity, 67
complexity added to application, 79
complexity and dictionary, 146
complexity and dictionary size, 147
complexity and increase ideas, 78
components of applications, 7
confused words, 115
confused words replacement, 105
contain bad words, 98
contain good words only, 98
contribution function, 63
contribution of people in application, 17
contribution people in project, 17
correct communication, 80
correct errors in communication, 63
corrected sentence, 96
correction error in communication reduce complexity, 74
Correction of Error in Application, 46
correction of the bad word, 51
correction process, 44, 52
Correction Process, 50
correctness application, 64

 Copyright 2023 *The Speak Logic Project*

corrects the underlying sentence, 44
create application/project, 62

D

definition already existed, 116
definition of words in other words, 106
degree of operation, 84
depend on our parent, 97, 114, 153, 157
depend on principles, 34
dependency on principle, 34
depends on communications, 59
derived entities, 130
derived word visibility, 132
derived words, 130
dictionary, 51
dictionary contains, 95
dictionary equal to parent dictionary, 99
dictionary equal to parent's dictionary, 136
dictionary increase from philosophy, 157
dictionary looks like parent's dictionary, 99
dictionary needed, 135
dictionary produced from parent dictionary, 99
dictionary serves no purpose, 102
dictionary size reduction, 139
dictionary structure, 95
dictionary with no negative words, 96
discard unwanted ideas, 82
disregards group of good words, 48
drive application, 68
driven by a single sentence, 78
driven by communications, 7
drop words that identify no entity, 107
dropped words from existing structure, 106
dropped words from existing words, 106

E

each person in application applies the principle, 62
ECF and reducing complexity, 146
effect of complexity in application, 67
eliminate bad words, 114
eliminate bad words from dictionary, 98
eliminate extra words, 139
elimination process, 114
employee cannot communicate for another, 57
employee has problem statement, 66
employee in application has function, 57
employee outside the application, 57
employees not physically present in application, 23
enable communication, 155
enable communication to be portable, 102
entities and size of dictionary, 130
entities that make up problem statement, 8
entity existence, 131
entity points to other entity, 128
entity produce, 26
entity referred by words, 107
entity usage and entity produced, 26
equal size of dictionary, 134
equal to parent dictionary, 105
error correction, 42
error correction function, 44, 45, 48, 51
Error Correction Function, 146
Error Correction Function existence, 156
error correction function is natural, 45
Error Correction Function removes complexity, 73
error free communication, 45
error free sentence, 49
error identification, 43
error in communication add complexity, 72
errors in communication, 42
examining incorrect sentence, 51
exchange information about application, 33
excluding principle, 50
execute a function to solve that problem as a second entity, 54
execute properly, 60
execute without error, 52
executed in area, 61
execution of application without error, 44
exist at least one good word, 52
exist similarity between words, 51
existence entity enables existence word, 122
existence function verification, 64
existence of communication, 100
existence of entity, 112
existence of feedback, 155

 Copyright 2023 *The Speak Logic Project*

existence of information, 100
External Communication, 24
external function, 58, 60
External Function, 56
external function dependency, 59, 60
extra communication, 76, 79, 87, 90
extra communication and complexity, 150
extra communication enables extra idea, 72
extra communication increases complexity, 72
extra communications, 71
extra communications not enable application, 71
extra idea, 75
extra idea increases complexity, 72
extra ideas with extra communication, 71
extra resources, 87
extra words, 136

F

feedback alert, 155
feedback exists to enable correction, 155
feedback process, 94
feedback reduce complexity, 74
feel stable about communication, 156
fewer relationship, 67, 89
first bard word, 51
first identify the problem, 53, 55
first step, 54
first we identify the problem, 54
flow of communication, 11
flow of communication in project, 11
forced function increases complexity, 85
function depend on other function, 59
Function Dependency, 59
function driven by communication, 70
function Execution Location, 49
function existence, 132
function is executed in relationship . function that solves the problem as third entity, 54
function management, 56
function no longer needed, 85
function not assigned, 58
Function of Application, 10
function of application verification, 64
function of commination, 59
function of communication, 11, 56, 57, 60, 61, 62
function of manager, 57
function of people in application, 12
Function of People in Project, 15
function of word, 125
function time line, 17
function verification in relationship with problem statement, 65
functions as entities, 132
functions dependencies, 59
functions derived from entity, 132
functions of communication, 59
functions of people in application, 57
Functions Related to Locations, 61

G

good idea, 81
good word from the dictionary, 52
good words, 47, 51, 114
good words dictionary exist only no bad words, 103
group ideas of application, 25
group of bad words, 47, 51, 52
group of good words, 47, 52
group of people managed, 56
group of sentences, 111
group of words, 45, 50
group of words with errors, 50
group of words without error, 50
group project's communication, 14
groups of words, 47

H

hanging the context of sentence, 44
help solving problem, 135
highest degree of operation, 84

I

I perform analysis of communication, 30
idea of lemon, 25
idea of lemonade, 25
idea of providing sugar, 25
idea of sugar, 25

 Copyright 2023 *The Speak Logic Project*

identify a sentence with error, 44
identify a solution for the problem, 55
identify actual entities, 121
identify actual entity, 117
identify entity word points to, 52
identify function in application, 63
identify principle of communication, 29
identify principles, 32
identify principles in communications, 33
identify resources, 91
identify similar words, 52
identify the problem as the first entity, 54
identify the problem first, 53
Identifying Error in Application, 42
Identifying the Principle, 29
implemented in other words, 116
improperly identify, 131
incomparable word, 123
incorrect communication, 79
incorrect idea, 79
incorrect sentence, 51, 52
increase idea of application, 78
increase number of communication, 79
increase number of ideas, 79
increasing relationships, 67
Individual Action, 19
Individual Problem Statement, 19
Individual Reason, 19
information about entity, 113
information and size of dictionary, 130
information as a single sentence, 155
information enables communication, 100
information enables comparison, 124
information existence, 155
information exists first, 100
information in relationship, 155
information is natural, 155
information principle as principle of operation, 34
Information Principles, 33
instant words removal, 139
intended word, 118, 119
internal function dependency, 59, 60

L

lacks of the principle of communication, 50
later time, 140
lemonade application, 8
less complex, 74, 89
list of information principles, 34
list of unneeded words, 140

M

M.s Smith's function validated by the principle, 63
main communication, 14
main entity visibility, 132
main function, 58
main function existence, 133
main idea of application, 25
main problem, 20
Manage Employee, 56, 57
manage size of dictionary, 130
management, 56
manager, 56
manager adds function for employee, 66
Manager assigns *Functions to employees*, 57
manager has problem statement, 66
manager of project, 56
manager uses communication to manage functions of employees, 56
Mary as communication holder, 22
Mary's idea, 24
Mary's function, 23, 24
Mary's idea, 25
Mary's Location, 49
match our parent dictionary, 97
match parent dictionary, 97
me and communication relationship, 36
me, communication, and portability relationship, 39
minimal communication, 80
minimal idea, 80
minimal work, 81
misidentifying problem, 92
misinterpreted word, 119
misinterpreted words, 109
misunderstood word, 119
mobile, 61
more effort, 89
more effort from us, 67
more relationship, 67, 89

 Copyright 2023 *The Speak Logic Project*

moved, 61
Ms. Smith analyzes, 30
Ms. Smith performs analysis, 29
Ms. Smith replies after analyzing communication, 30
Ms. Smith's communication, 14, 22, 63
Ms. Smith's function, 16, 23
Ms. Smith's location, 61
Ms. Smith's Action, 20
Ms. Smith's communication, 10, 59
Ms. Smith's idea, 25
Ms. Smith's Location, 49
Ms. Smith's problem statement, 19, 20
multiple aspects relationships, 41
multiple communication in relationship with multiple ideas, 70
multiple ideas, 69
multiple parts, 21
My Action, 20
my communication, 10, 14, 22, 63
my function, 23
My function, 16
my function validated by the principle, 63
my idea, 25
my location, 49, 61

N

natural communication, 71, 144
natural communication enables application, 71
natural communication reduce complexity, 73
natural entities, 77
natural idea, 77, 81
natural idea of application, 78
natural sentence, 148
natural to application, 146
naturally driven application, 145
naturally needs, 88
need correction, 156
need of feedback, 155
need other application, 60
needed resources, 87
negative functions increase complexity, 85
negative word, 131
negative word improper entity, 131
new words, 103
no feedback or alert, 155
noncomplex entity, 67
not appear in model, 59
not feedback needed, 156
not in our model, 60
not physically presents in project, 23

O

occupy more space, 122
on top of everybody, 28
one cannot apply principle for each other, 57
one cannot apply the principle for each other, 57
one cannot assume each other's functions, 59
one cannot communicate for each other, 59
one good word, 52
only people in the project can execute functions of the project, 57
operating principle, 28
operating principle common among people, 84
operating principle reduce complexity, 84
opposite word, 118
original dictionary, 136
our dictionary, 96
our Function in our Application, 62
our Function in project, 62
outside communication, 76
outsider cannot come to project, 57
overall communication, 22
overall function validated by the principle, 63
Overall Project Communication, 14
overall project's communication, 15

P

parent alert, 156
parent dictionary, 94
parent dictionary produces our dictionary, 99
parent dictionary structure, 95
parent hear children, 155
parent still corrects our communication, 143
parent vocabulary, 94, 95

 Copyright 2023 *The Speak Logic Project*

parent vocabulary less, 94
part of application in relationship people and communications, 21
partition dictionary, 98
Partition of Application, 21
Partition of Project's Communication, 22
parts of entity, 127
parts of entity from main entity, 130
parts of entity point to entity, 127
parts of entity visibility, 132
passing information, 100
People in Application, 12
people in application and complexity, 75
people in application and their communications, 22
people in application depend on same principle, 84
people in project depend on principle of communication, 57
people outside, 58
people work together, 17
people's communication and complexity, 75
perform analysis, 51
perform replacement, 51
performance of our application, 82
performs the correction, 48
performs the correction on the group of bad words, 48
person from Area 1 added to application, 65
person outside the project, 57
person outside the project cannot be a part of the project, 57
Peter’s communication, 14, 22, 63
Peter’s Function, 16
Peter’s function validated by the principle, 63
Peter’s location, 61
Peter’s problem statement, 20
Peter's Action, 20
Peter's communication, 10, 59
Peter's function, 23
Peter's idea, 25
Peter's Location, 49
Peter's problem statement, 19
philosophy inheritance, 141, 157
philosophy inheritance enables dictionary increases, 157
pick a good word, 52
pick that good word, 52
point nowhere, 122
portability and principle relationship, 36
portability of principle, 36
portability relationship, 37
portable dictionary, 99
portable sentence, 52
present information properly, 101
previous negative, 103
principle common among, 84
Principle Number, 32, 33, 52
principle of communication enables application, 28
principle of communication includes in communication, 83
principle of operation, 28, 34
principle of operation model, 29
principle on top, 28
principle on top of everybody, 28
principle relationship, 83
Principle Verification, 35
Principles, 32, 33, 52
principles as single entity, 35
Problem Action, 8
Problem and Solution, 53
problem existed by word, 118
problem exists solved by parent dictionary, 138
problem identification, 54, 92
problem identification process, 54
problem identified and complexity, 92
problem in area, 62
Problem Reason, 8
problem remains unsolved, 92
problem replacement, 128
problem solved, 53
problem solved by function execution, 53
problem solved by words, 118
problem solving and dictionary size, 103
problem statement, 8, 53, 65
Problem Statement Elements, 8
Problem Statement Entities, 8
problem statement identified, 8

 Copyright 2023 *The Speak Logic Project*

problem statement reasons, 20
Problem Statement's Action, 8
problem statement's reason, 19
Problem Statement's Reason, 8
process of elimination, 114
process of solving problem, 53
produces corrected sentence, 48
produces good word, 51
project's problem statement, 19
promote instability, 123
promote stability, 122, 157
provide feedback, 60
provide function description, 57
provide information properly, 101
purpose of dictionary, 102
purpose of vocabulary, 102

R

Reason, 8
Reason Taken Action, 8
reduce number of communication, 81
reduce number of ideas, 81
reduce size little by little, 139
reduce size of dictionary, 103
reduce size of our dictionary, 97
reduce the size of our dictionary, 97
reducing complexity, 81
reducing complicity steps, 92
reducing number of words, 103
reduction words, 96
refers to a group of word, 121
relationship between resource and communication, 88
relationship not hold, 52
relationship of the principle and our aspect, 40
relationship problem and function and solution, 55
remaining to be removed, 140
remove all negative words, 97
remove confusion, 116
remove error, 48
remove errors reduce complexity, 74, 81
remove ourselves from information, 101
remove unwanted communication, 82
remove words that identify no entity, 107
remove words with no structure, 107
removing words that identify no entity, 112
removing words without aspects, 113
repeat bad words, 95
replace bad words, 51
replaced with solution word, 129
replacement of the bad words, 44
replacing bad words with good words, 44, 48
represent a sentence, 45
requires execution of a function, 58
resource unallocated, 91
resources, 87
resources allocated, 91
resources communication, 87
resources identified, 91
resources natural to application, 87, 91
resources needed, 88
resources to solve problem, 92
resulted relationship, 37
resulted sentence, 49

S

second step, 54
secondly we identify the solution, 54
Secondly we identify the solution for the problem, 54
sections of communication, 7
segments of communication, 7
self-sufficient, 94, 97, 100, 114, 135, 153
sense of sentence, 52
sentence belongs to dictionary, 101
sentence is correct, 51
sentence natural to application, 78
sentence of words, 95
sentence with error, 46, 50
sentence with errors, 95
set of communication principles, 35
set of good words, 52
set of information principles, 35
set of sentences, 139
set of words, 51
set of words identify entities, 112
set of words identify no entities, 112
similarity between, 51
similarity between communication, 83

 Copyright 2023 *The Speak Logic Project*

similarity between the words, 52
single communication, 80
single idea, 77, 81
single sentence, 51
size of communication, 67
size of dictionary, 96
size of dictionary and problem solving, 103
size of dictionary equal to parent, 136
size of dictionary getting big, 103
size of dictionary needed, 100
size of dictionary related to entities, 130
size of dictionary related to information, 130
size of our vocabulary, 94
size of vocabulary and stability, 153
smaller size, 97
Solution, 53
solution exists, 118, 128
solution exists in other words, 118
solution for . problem, 55
solution identification, 54
solution identified, 53
solution process, 55
solution related to affected entity, 55
solution verification, 64
solution word, 118
solved problem, 64
solved the underlying problem, 78
specific word in group, 121
stable enough to communicate, 153
state and information, 122
state preserved by information, 122
step of solving problem, 54
structure already included, 116
structure included in other words, 106
structure not exist in combination, 118
structure of parent dictionary, 95
structure of words, 107
sub words, 132
subfunction, 58
subfunctions existence, 133
substitute word can solve problems, 128

T

term or an expression, 122
the main function of application, 56
timelines with communication holder, 23
too many words in dictionary, 101
total communication, 14
total project communication, 14
two groups, 50, 52
two steps, 54

U

unallocated resource, 91
unallocated resources, 91
underlying problem, 53
underlying solution, 55
unique entity, 123
unique word, 123
unnecessary application, 67
unnecessary words, 94
unneeded communication, 90
unneeded function increases complexity, 85
unneeded resources, 90
unpotable words, 95
unwanted communication, 74, 75, 76, 77
unwanted function increases complexity, 85
unwanted idea, 75
unwanted ideas, 77
unwanted resources, 90
unwanted words, 140
usage of the principle, 28
use the principle of communication, 52

V

valid function, 63
validated by the principle, 63
verified by the principle, 63
verify application, 64
verify application from problem statement, 65
verify function of application, 64
verify solution, 64
version of application, 89
visual aspect of communication, 91, 123
Visual Aspect of Communication, 110
visually identified entities, 110
vocabulary affects stability, 153
vocabulary needed, 135

 Copyright 2023 *The Speak Logic Project*

W

ways to reduce size of dictionary, 97
well interpreted replaced with, 120
well misunderstood word, 120
well understood words, 109
what we do, 18
what we do function, 18
what we do with overall communications, 18
word by word analysis, 51
word cannot solve problems, 128
word exist from entity, 122
word existence, 131
word identifies wrong entity, 120
word in the group, 121
word point to solution, 119
word points to wrong entity, 127
word promote stability, 122
word refers group, 110
word *solution*, 128
word that cannot identify entity, 112
word usage develop problems, 118
words and problems solved, 135
words belong to dictionary, 101
words belong to parent dictionary, 145, 155
words children use, 94
words derived from previous negative, 103
words develop problem remove, 108
words existence, 102
words from dictionary, 51
words have been misused, 116
words identify wrong entities, 109
words intended to solve problems replacement, 108
words natural to information, 155
words parent uses, 94
words point to same entity, 120
words point to wrong entities, 95
words reduction generation basis, 141
words that do not identify entities, 107
words to solve problems, 137
words used that don't exist, 102
work harder, 81, 89
work harder for nothing, 81
work together, 17
working area, 61
working area of people in application, 61
wrong context, 44
wrong words, 44
wrongly identifies *Entity*, 120

Copyright 2023 *The Speak Logic Project*