

# THE SAMPLING ANALYSIS PATTERN

H.A. Sánchez, Binbin Lai, and M.E. Fayad

Computer Engineering Dept., College of Engineering, San Jose State University  
One Washington Square, San Jose, CA 95192-0180  
huascar\_sanchez@yahoo.com, binbinlai@yahoo.com, and m.fayad@sjsu.edu

**Abstract**— Sampling is a general concept that has many applications in various domains. The idea of representing sampling as a pattern is to guarantee a reusable core. The stable sampling analysis pattern is introduced and defined as a solution for providing the core knowledge of the sampling problem itself. In order to achieve this goal, the Sampling pattern is built based on the software stability concepts approach introduced in [1]. The Software Stability Concepts provide the Sampling pattern a stable and reusable core [1]. This core is represented in terms of Enduring Business Themes and Business Objects artifacts [1]. Due to their reusable and stable nature, they grant the ability of this pattern to be used in other applications which share the same knowledge. This paper provides detailed documentation of the proposed -stable sampling analysis pattern.

**Index Terms**—Software stability, Software patterns.

## I. INTRODUCTION

When referring to the term sampling, we are entering into a multi-application area of study. This term as an action is applied in almost every activity in our daily lives. Sampling utilization ranges from small and simple activities, such as sampling a small portion of a cake at the supermarket to taste the cake, to the most complex ones, such as the action of determining the percentage of contamination occurrence in hard-drives manufactured in the month of July 2003 at the Seagate Company. Due to the impossibility of studying large volumes of the population, researchers rely constantly on sampling to single out small portions of a particular population to perform an experiment or evaluation study. For instance, in performing an investigation of a customer's satisfaction against a store's service. Since it would be impossible to ask every customer for their opinion, the store would have to make use of sampling to randomly choose customers for the investigation. Similarly, the research of the use of certain Educational Software might involve the sampling of picking 20 students out of a total of 500 students. All given examples provide strong evidence of the utilization of sampling across different domains; explicitly exhibiting the

true reusable nature of the Sampling Analysis Pattern across many domains.

Sampling as a concept is a very general term that can span multiple applications. It is defined as a technique used to capture continuous phenomena from a universe, providing an idea or estimation of that particular universe [4]. There are different kinds of sampling techniques that are employed these days. For Simplicity purposes, only a few of them will be mentioned in this paper, such as Random Sampling, Cluster Sampling, Stratified Sampling, and Quota Sampling.

*Random sampling* is a sampling technique where a group of subjects are selected for a study to represent a larger group of the population. Each subject is randomly chosen. Each of these members or subjects that are part of a particular population has an equal chance of being included in the sample. Every possible sample of a given size has the same chance of selection [5]. *Cluster Sampling* is a sampling technique where the entire population is partitioned into groups. Then, a random sample of these clusters is selected. All observations in the selected clusters are included in the sample. Cluster Sampling is usually used when the researcher cannot get a complete list of the members of a particular population they wish to study, but can get a complete list of clusters of the population [5]. *Stratified Sampling* is a technique that takes samples from each stratum or sub-group of a population [5]. This is driven by the occurrence of factors that partition a population into sub-populations or sub-strata. *Quota Sampling* is a technique that usually applies in market research and opinion polling [5]. A person in charge of sampling is given a quota of subjects of a specified type to attempt to record a certain phenomena and perform a specific action (e.g. interviewing).

These Sampling techniques are employed usually to cope with a particular problem domain. Concretely speaking, they are structured in such a manner that is solely focused on a solution for a specific problem. In the case that they are to be implemented in a distinct domain, in which elements are different in characteristics and behavior, it is possible to obtain an inaccurate result; and hence, a failure of the sampling action. Therefore, being able to address all the different varieties of problem domain solutions, where the Sampling is

incurred into one core abstraction, is a challenging a valuable task.

The intent of this paper is to extract the core insight of the sampling term, and to represent it as a stable pattern to serve as a stable and reusable core for other applications sharing the same core domain. In order to achieve this goal, the Sampling pattern is built based on the software stability concepts [1]. The Software Stability concepts provide a stable and *reusable* core for multiple applications sharing the same core insight or knowledge [1]. In our case it is the Sampling core knowledge. Software Stability concepts will partition the Sampling term into EBTs, and BOs. These artifacts, due to its stable and *reusable* nature, can represent the basis for patterns definition. For further information on how to identify patterns using Software Stability Concepts, please refer to [1].

The section below provides a detailed description of the Sampling Analysis Pattern. This pattern presents the atomic core of the sampling definition itself based on the Software Stability Concepts (SSM) paradigm. The idea of applying SSM in the discovery of an atomic abstraction of the problem "Sampling," is to guarantee stability and reusability; and therefore, this pattern can be used to model the same problem whenever it appears.

## II. PATTERN DEFINITION

### **Pattern Name: *Sampling Pattern***

This pattern represents the process of selecting a small portion or piece of items as a sample to represent a larger item or group of items.

Current implementations of sampling and its kinds are usually bound to a specific problem domain. Each technique's structure is constrained to provide solutions to a fixed problem domain. This makes them unsuitable for usage across other problem domains. Such fallout explicitly restricts the underlying capability of Sampling to be applied in a vast number of applications. Sampling is a widely used term that has several built-in essential characteristics, such as its capacity to cover multiple areas of application, its ability to enclosed distinct selection methods within its core mechanism for sampling, and the sampling capability itself which provides the base for today's used sampling techniques. Such characteristics are exhibited within the Sampling Pattern model definition. There are several aspects of the Sampling pattern definition that transcend in all the distinct areas where sampling is applied. For example, it generalizes all the sampling problem solutions into a unique one that is applied across multiple domains. Also, it allows the separation of the problem into different categories (Enduring Business Themes, and Business Objects). These categories possess a stable and *reusable* nature. Therefore, they will form the basis for building as many applications as you want with stability and reusability in mind.

### **Context:**

In our daily lives, there are many situations where sampling takes place. For example, knowing the impact of advertisements in specific areas involves some sort of sampling. In Stock market research, sampling seems to appear constantly. For instance, knowing if is worthy to invest in a particular company based on the company's previous background. This might involve a sampling of the up and downs of the company in determined period of time.

Generally speaking, Sampling is an essential term widely used in different domains. Its core definition enclosed the capacity of handling different types of populations relying on specific criteria, different methods used to study those populations, etc. Therefore, defining a pattern, which structure captures the core abstraction of the Sampling definition itself, is a valuable and challenging task. However, this abstraction is not easy when trying to handle all the distinct types of sampling problems in one pattern. This pattern will embody the sampling concept itself. Its utilization will be spanned by any application domain that involves a sampling necessity. This *reusable* characteristic will be accomplished by using the Software Stability Concepts throughout the entire pattern definition process.

### **Forces:**

The Sampling Analysis Pattern should resolve the following forces:

- This Sampling process spans multiple applications different in nature. Therefore, this pattern should be general enough to capture the core knowledge of the sampling process itself, and then to handle these heterogeneous applications through its stable and reusable capabilities.
- The Pattern should embody the different sampling categories or types. Current solutions are solely focused on a specific category.
- Sometimes the sampling process can be conducted over a large number of different Medias, either simultaneously or consecutively. Therefore, this pattern should be flexible enough to embody the distinct types of Medias within its structure.
- Sampling can be performed by one or multiple entities at the same time. These entities are distinct in nature and they can be represented by one or a group of persons, organizations, companies, etc. These entities are capable of playing distinct roles in a sampling process. For instance, there can be entities defining a sample from a particular population of items, others who are specifying the criteria for the analysis, etc. Therefore, our pattern should be stable and flexible enough to handle a variety of structures and roles from its performers.
- The pattern needs to be abstract enough and not tied to one specific mechanism in order to accomplish a particular result. These mechanisms could vary in nature, and in the approach utilized to perform the

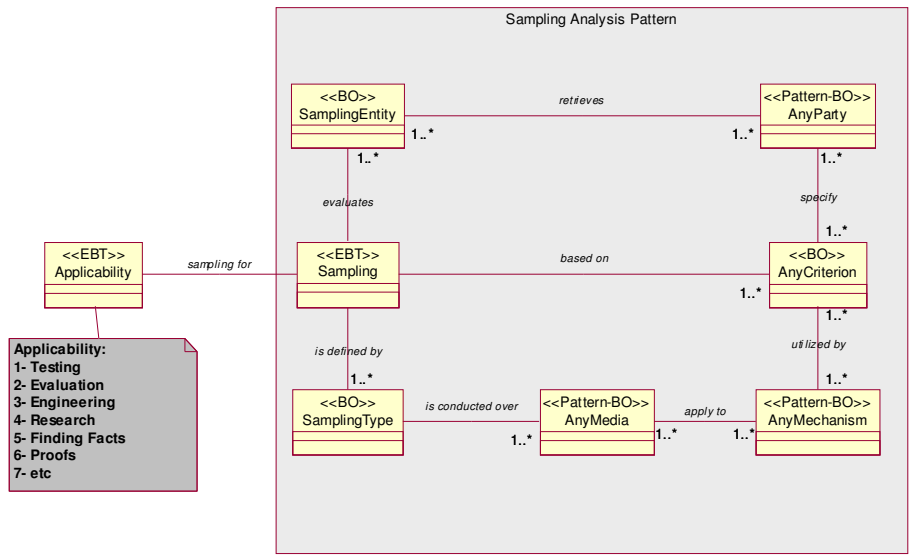
sampling process. Our pattern should exhibit a great flexibility and abstraction in order to handle these countless mechanisms.

- The pattern should be flexible enough to allow the execution of these mechanisms in parallel or in sequence.
- The mechanisms used during the Sampling process are nourished by certain criteria used to determine the sampling results. These criteria can represent different and countless parameters that initialize the sampling process. Also, these criteria can be specified by one or more persons, organizations, companies, etc. Therefore, our pattern should be flexible enough to handle not only the different criteria, but also all the distinct performers.
- The pattern needs to be abstract enough to cope with different types of sampling entities. Sampling Entities embody those representative items from an entire set or group of entities. For example, a blood sample, or urban area sample, etc.
- Sampling can be performed on one or more populations at the same time. For instance, sampling that takes place in the quality control aspect usually involves more than one subject to be sampled. For example, in finding out the defect rates of a product, one can sample the length, width, height, resistance, and so on. Therefore, the pattern should be flexible enough to handle such situations.
- This pattern should exhibit a great compatibility with the distinct areas of application where its use is requested. That is, these areas vary in characteristic, constraints and purposes. Therefore, this pattern needs to be stable and abstract enough to encompass and efficiently handle these issues.

The ultimate goal of any sampling process is to select a sample from a given population of items. However, the nature of this selection varies tremendously from one application to another. This selection is driven by the criteria or parameters specified by the Performer of the Sampling Process. For example, the selection of a sample from a blood test is completely different than the sample of the machine parts. Therefore, this pattern should be flexible enough to offer an effective parsing of these criteria into the selection process.

### III. PATTERN SOLUTION

The following model will represent the proposed solution of the Sampling Analysis pattern, using the Software Stability Concepts approach. See Figure 1:



**Figure 1:** Sampling Pattern stable object model.

**Participants:**

The participants of the Sampling Analysis Pattern are:

**CLASSES:**

**Sampling:** Represents the Sampling process itself. This class contains the characteristics and behavior that initialize the sampling process.

**SamplingType:** Represents the type of sampling to be executed at certain time. Before starting any sampling process this is the class that will be requested.

**SamplingEntity:** Represents the population to be sampled. It could be considered as the fuel that feeds the sampling process. This presents a core element in any sampling process. If more than one responsibility is identified, additional classes should be formed. Limiting responsibilities will help prevent low cohesion and high coupling as well as reduce the possibility of macho classes.

**AnyCriterion:** Represents the parameters the sampling process will be running on.

**Applicability:** Represents those areas where the pattern Sampling, due to its stable and reusable nature, can be used. This class will embody the extended boundaries of the Sampling Pattern.

**PATTERNS:**

**AnyMechanism:** Represents the mechanisms that will be used by distinct media to conduct the sampling process. It models all the methods that are involved in the sampling process.

**AnyMedia:** Represents the media through which the sampling process will take place. For instance, one can sample the occurrence of winning numbers of the Lotto in the year 2000 by accessing the Lottery’s website over the Internet. Others might use the newspaper or the TV to sample this occurrence pattern. The pattern diagram and detailed pattern description is provided in [2].

**AnyParty:** Represents the sampling inducers. It models all the parties that are involved in the sampling process. A Party can be a person, organization, or a group with a specific orientation. The pattern diagram and detailed pattern description is provided in [2].

**CRC Cards:**

The CRC card names the class, responsibility, and its collaborations. The CRC card also names a role for each class, which is useful for identifying the class responsibility. Each class should have only one and unique responsibility. The collaboration consists of two parts: clients and server. Clients are classes that collaborate and have relationship with the named class. The Server contains all the services that are provided by the named class to its own clients [2]. A group of CRC Cards representing the Sampling Pattern is showed in Figure 2. Figure 3 shows the Sequence Diagram of the Sampling Analysis Pattern.

Sampling (Sampling Handler)		
Responsibility	Collaboration	
Describes the sampling concept itself.	<b>Clients</b>	<b>Server</b>
	SamplingEntity SamplingType AnyCriterion Applicability	defineProperties() specifyScope()

SamplingEntity (Population Descriptor)		
Responsibility	Collaboration	
Describes the entities used for the sampling process.	<b>Clients</b>	<b>Server</b>
	Sampling AnyParty	retrieveBehavior () detailProperties() knowBackground()

AnyMechanism (Method Descriptor)		
Responsibility	Collaboration	
Represents the abstract mechanisms used for the media to assist the sampling process.	<b>Clients</b>	<b>Server</b>
	AnyCriterion AnyMedia	requestCriteria () runMethod() methodConstraints() integrateCriteria()

AnyParty (Sampling Inducer)		
Responsibility	Collaboration	
Induce a sampling process over a particular population.	<b>Clients</b>	<b>Server</b>
	SamplingEntity	sample () monitorSampling() stopSampling()

Applicability (Applicability Descriptor)		
Responsibility	Collaboration	
Defines the applicability of using the pattern.	<b>Clients</b>	<b>Server</b>
	Sampling	chooseApplication () apply ()

SamplingType (Sampling Identifier)		
Responsibility	Collaboration	
Identify the type of sampling method.	<b>Clients</b>	<b>Server</b>
	Sampling AnyMedia	representType () associateMethod() requireApproach ()

AnyCriterion (Criterion Descriptor)		
Responsibility	Collaboration	
Embody the set of parameters or criteria used to initialize the sampling process.	<b>Clients</b>	<b>Server</b>
	Sampling AnyParty AnyMechanism	establishParameter () identifyDomain() exhibitProperties ()

AnyMedia (Media)		
Responsibility	Collaboration	
Represent the Media over the Sampling Entities would be sampled.	<b>Clients</b>	<b>Server</b>
	AnyMechanism SamplingEntity	mediaCapability () illustrate() nameMedia()

AnyParty (Criteria Handler)		
Responsibility	Collaboration	
Specify certain criteria to invoke a particular mechanism of the sampling process.	<b>Clients</b>	<b>Server</b>
	AnyCriterion	fillCriteria () initializeMethod() editCriteria() dropMethod()

Figure 2: CRC Cards representation of the Sampling Analysis Pattern

SAMPLING PATTERN - SEQUENCE DIAGRAM:

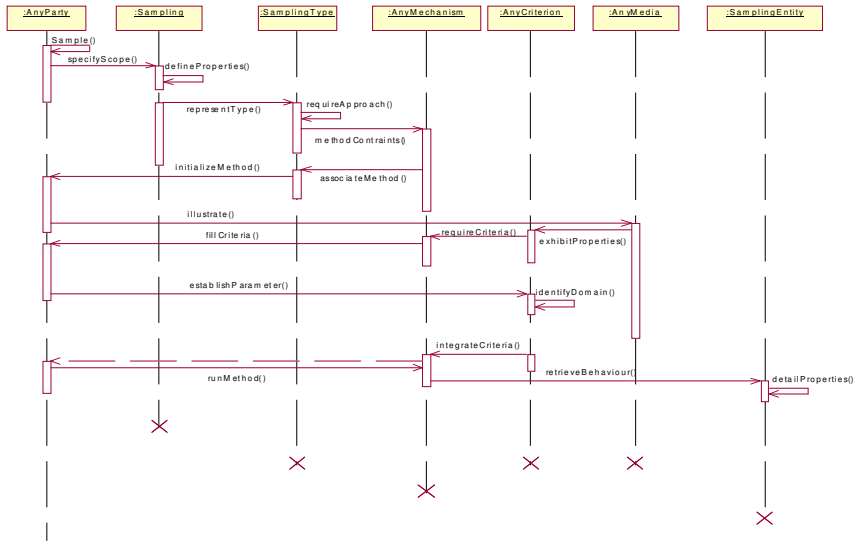


Figure 3: Sequence Diagram with Stability in Mind

**Consequences:**

- The use of the Sampling pattern offers the following benefits:

1.- *Handling more than one Population:* Unlike current solutions for a sampling problem, where a new model is generated per population at a time, the Stable Sampling pattern does consider the situation of having more than one population within the same application. This is done through the use of the SamplingEntity Business Object.

2.- *Embody Different Sampling Types:* The Sampling pattern is abstract enough to embody the plethora of sampling types used for distinct types of problems. Such capability is done by providing the core abstractions of these types within the SamplingType Business Object.V. Applicability of the Proposed Format

3- *Handle Different Mechanism:* The Stable Sampling pattern is general enough to handle distinct mechanisms for sampling, different in nature and process flow. With such variety of mechanism, the sampling pattern needs to adjust its properties in order to accomplish a proper sampling action. This is done by the use of the AnyMechanism pattern.

4- *Consider Different Media Types:* The Stable Sampling pattern considers the utilization of certain mechanisms over different media types. This is accomplished by using the AnyMedia pattern, which represent the media type and its kinds. This feature increases the flexibility of the pattern since the sampling problem is needed in different applications through the use of different media types, such as the Internet media, TV media, Poll media, etc.

5- *Adaptable for Required Application Areas:* The Stable Sampling pattern structure maintains a high level of adaptability across different application areas. This sampling pattern represents a stable and *reusable* core that can be utilized in different areas that share the same problem domain. The determination of these application areas will adapt the sampling pattern to best meet the goals of this sampling concern.

- The use of the Stable Sampling pattern has the following limitations:

1- Lack of Pattern Representation. At first impression, it would be hard to discover, in a wide sense, the several hidden

concerns within the patterns that are included in the stable Sampling pattern. Such as, concerns related to the assignment of roles of the entities performing a sampling problem; however, these concerns should be considered within the AnyParty pattern details.

2- No Industrial Objects to Clarify Pattern Applicability. Since the Stable Sampling has been developed based on software stability concepts, there are no IOs attached to the pattern itself, which makes the pattern's applicability not very obvious from just reading the Sampling pattern structure. However, attaching such IOs (which are implementation details) will narrow the applicability of the pattern. Showing detailed case studies for the pattern applicability make the pattern usage obvious; yet, preserve the generality of the main pattern.

#### IV. PATTERN APPLICABILITY

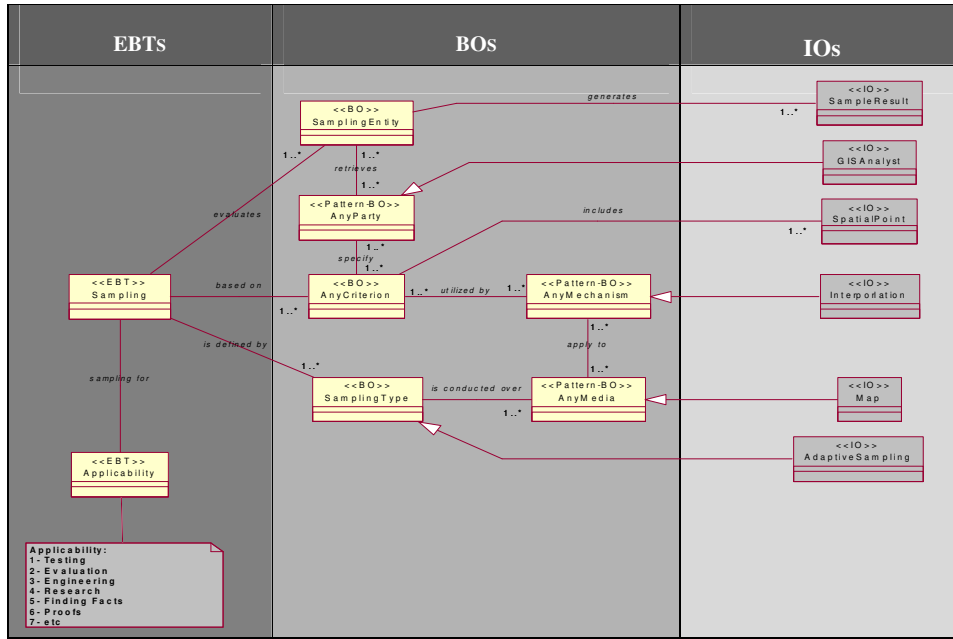
In order to illustrate the use of the Sampling pattern in different application areas, two examples are presented: Sampling unknown regions for GIS research using Adaptive Sampling, and Cluster Sampling to verify Increasing Volumes of Data in Database Systems. Since the purpose of these examples is to demonstrate the usage of the proposed pattern, and for simplicity, these examples do not present the complete model for the problem. Instead, they focus on the part that involves the sampling process.

The following examples illustrate the use of Sampling in different applications.

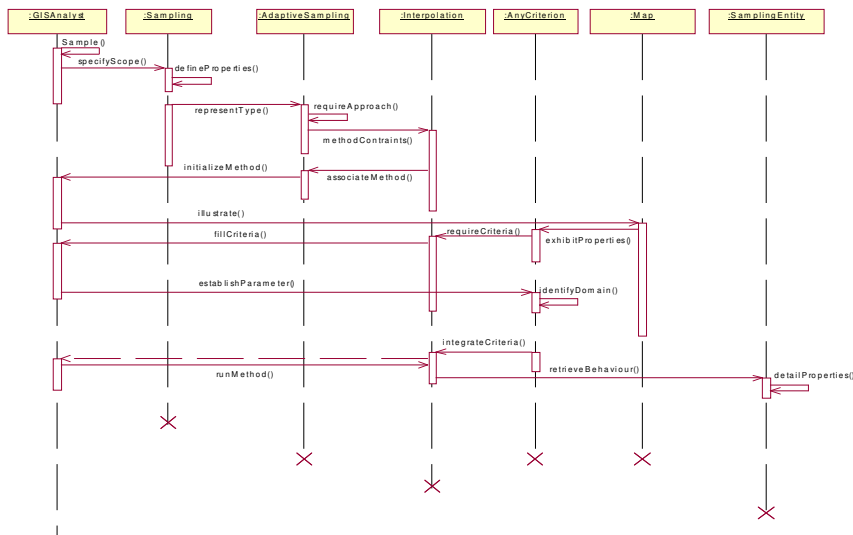
##### **Example 1: Sampling Unknown regions for GIS Research using Adaptive Sampling:**

Geographic Information Systems require data at all points of distinct geographic regions. However, it is almost impossible to measure an infinite series of points in a determine plan. This example models a simple solution to retrieve certain points from an unknown region using adaptive sampling techniques. Figure 4 shows the stability model of the sampling used in *GIS Research*. Classes that are not in the original *Sampling* pattern are colored in gray. Figure 5 shows the Sequence Diagram for this solution based on Stability.

**Comment [W1]:** What is a determine plan?



**Figure 4:** Stability Model of the Adaptive Sampling solution for GIS Research.



**Figure 5:** Sequence Diagram using Stability in Mind

**Example 2: Cluster Sampling to Access Increasing Data Volumes in DB Systems.**

Today, the necessity for users to determine accurate results from large volumes of data in Database System has become a problem on a daily basis. The example models

an approach used by IBM Researchers to process less data, and approximate results using Sampling Techniques [3]. Figure 6 shows the stability model of the sampling problem

in Database Systems. Classes that are not in the original Sampling pattern are colored in gray. Figure 7 shows the Sequence Diagram for this solution based on Stability.

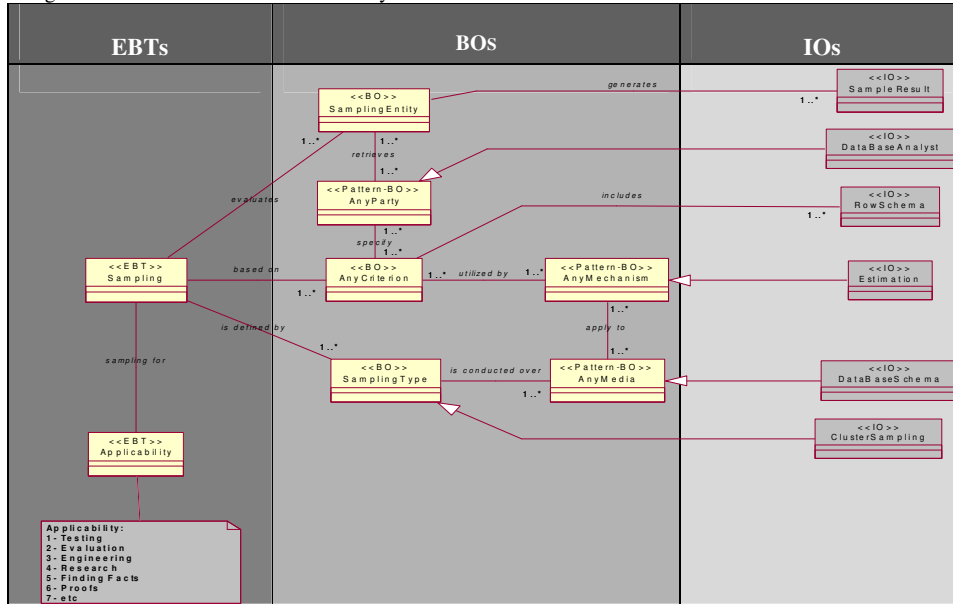
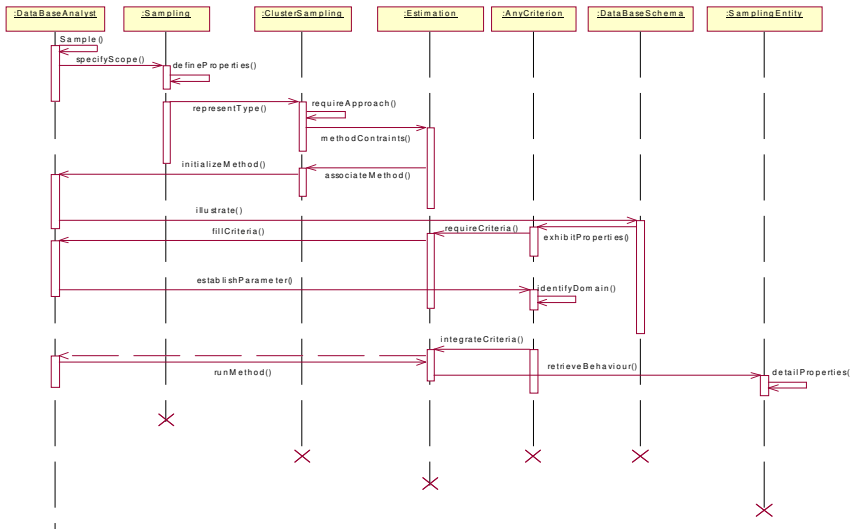


Figure 6: Stability Model of the Cluster Sampling solution for Database Systems.



## Figure 7: Sequence Diagram with Stability in Mind.

### V. CONCLUSION

The main objective of the work described in this paper is the utilization of the Software Analysis concepts to an application neutral Sampling pattern. The implementation of this objective has resulted in a stable and reusable solution for a countless number of applications sharing the same knowledge of a sampling action. One of the main contributions of this work is the identification and modeling of an atomic Sampling term as a pattern to serve as stable and reusable core. There are many reasons why our pattern is considered robust and valid for posterior use. One of these reasons is a clear separation of concerns. This is done by separating the core abstractions of the problem using Enduring Business Themes, and Business Objects from the changeable artifacts which represent the actual implementation of the application. Second is its capacity of being reusable, customizable, traceable, and adaptable across multiple problem domains through the special built-in characteristics of EBTs and BOs.

### REFERENCES

- [1] M.E. Fayad. "Accomplishing Software Stability." Communications of the ACM, Vol. 45, No. 1, January 2002.
- [2] M.E. Fayad, V. Stanton, and Hamza, H. "A New Look At the CRC Cards."  
<http://www.activeframeworks.com>
- [3] Peter J. Haas, "Speeding Up DB2 Using Sampling", IBM Data Management Technical Conference, Anaheim CA 2002.
- [4] Tony Dent – Chairman, Sample Answers Ltd, "Probably the Best Sample You can Get", ASC Conference, Imperial College, 17<sup>th</sup> April, 2002.
- [5] Valerie J. Easton and John H. McColl, "Statistics Glossary v1.1".