**REPORT ON ANALYSIS PHASE LEARNING OUTCOMES**

**IT432- SYSTEMS ANALYSIS AND DESIGN**

**Introduction**

A project team may work on an IT project where either the final deliverables does not meet the client’s expectations or the project fails. In many cases these are as a result of classic mistakes that are related to people and processes. In order to minimize such occurrences, there is the need to break down the deliverables in the high-level project charter into the more detailed business requirements. This is done in the analysis phase of the SDLC. Requirements modeling, data (domain) modeling, and process modeling constitute the three major activities in the analysis phase and this class seek to have a deep dive into these major activities.

To better understand the three major activities in the analysis phase, the IT 432 class took several hands-on activities (Labs) in and outside of classes in the areas of domain model class diagram, concept map, and activity diagram. The purpose of this report is to give a detailed description about the learning outcomes of the analysis phase of SDLC covering the domain model class diagram, concept map, and activity diagram presented in class. Section one of this report focuses on domain model class diagram of the AI experiment “What we eat”. This diagram shows the things that the AI system will keep track of in order to function when working with the said experiment. Section two focuses on the concept map which depicts suggested relationships between the concepts studied in this class. Section three also focuses on the activity diagram which shows the steps required by Wolfram Alpha’s system (and modules inside it) to deliver the given feature of currency conversion to a group of actors. Below are the detailed descriptions of the three diagrams.

**Section 1**

**Domain Model Class Diagram of the AI Experiment “What We Eat”**

The first diagram is the domain model class diagram of the AI experiment “What we eat” of Lab 2 in this class. This diagram is used to show the things that the AI system will keep track of in order to function when playing the game. It contains the classes, attributes, object, associations, multiplicity, and multiplicity constraints. The classes are the category of things that the system keeps track of, and there are six classes in this class diagram with each class uniquely identified by its name written on top of the class. They are Player, Diet, Ingredient, Quality, RecipeYield, and QualityOfYield.

The attributes of each class are the set of features that describe the classes, and each attribute has a return type in which it is represented in the class. Examples include the Player class which has the attributes Gender which is of the type Boolean; the attribute Age which is of the type Integer; and the attribute Location which is of the type String. Objects are instances of the classes, and associations are the relationship among instances of classes. In this class diagram, the Player class may have one-to-one association with the Diet class; the Diet class may have one-to-one or one-to-many association with the Ingredient class; the Ingredient class may have many-to-many association with the Quantity class; and the Quantity class may have many-to-many association with the RecipeYield class. A solid line connecting any two classes in the diagram shows associate between the two classes. The QualityOfYield class is an association class because it has the attribute “DietTaste” that need to be remembered, and we use the dotted line to connect this class to the two classes Quantity and RecipeYield which have many-to-may associations between them.

A multiplicity is used in this class diagram to show the number of instances of a relationship that instances of the classes on either side of the relationship can be involved in. Also, multiplicity constraints are used to tells if there is an upper limit to the number of instances of the relationships for objects on either side. Examples include, a Player may have one and only one (mandatory) Diet represented as 1, but a Diet may have no player or one player (optional) represented by 0..1. A Quantity may have one or more (mandatory) RecipeYield and is represented by 1..\*, and a RecipeYield may have one or more (mandatory) Quantity which is also represented by 1..\*. These and others shown in this class diagram are the multiplicity and multiplicity constraints showing the details of instances of the relationship that instances of the classes on either side of the relationship can be involved in and their limits.

This domain model class diagram provides a basic notation for other structure diagrams prescribed by UML and is helpful for developers. Business Analysts can use this class diagrams to model the system from a business perspective.

**Section 2**

**Concept Mapp**

This section explains the concept map diagram which was presented in class. This diagram depicts suggested relationships between the concepts in SDLC studied so far in this class. Here, eight concepts were picked and were connected with links labeled that describe the nature of the connection.

From this diagram, association classes are mapped to M-to-M associations and is used to describe the relationship among instances of classes. This implies that a class may have many-to-many associations with another class. Example, the Quantity and RecipeYield classes have many-to-many association in the domain model class diagram of the AI experiment described in section one of this report.

Also, class is mapped to attributes and is used to show the category of things that the system keeps track of. This implies that a class may have one or more attributes within the class. Example, the Ingredient class in the domain model class diagram of the AI experiment described in section one above has Ingredient Name, Ingredient Category, and Color Additive as its attributes.

Further, activity diagram is mapped to use case, and is used to shows the step required by the system (and modules inside it) to deliver a given feature to a group of actors. This implies that an activity diagram in the use case module illustrates the flow of events of a use case and describes what needs to be done by the system to provide value to an actor.

Furthermore, domain model class diagram is mapped to relational database, and is used to show the relationships among instances of classes and things that the system will track of in order to function. Example, database columns are modeled as attributes of the classes in the class diagram and an attribute of a class may be defined as the primary key in the relational database system.

**Section 3**

**Activity Diagram of Currency Conversion on the Wolfram Alpha System.**

Section three focuses on explaining the activity diagram used to show the step required by the Wolfram Alpha’s system (and modules inside it) to deliver the given feature of currency conversion to an actor or a group of actors. In this diagram, there are two activity partitions User and Wolfram Alpha. The user starts by inputting a question “How much is $1000 in euros ?” The system evaluates the user’s question and displays the necessary inputs to the user. The user then chooses the currency to be converted and the currency to be converted into, and the submit to the system. The system receives and process the user’s request and displays an input menu for the user to enter the amount to be converted. The user then enters the amount and submit to the system. The system processes the data and display the converted amount in euros to the user, then the activity ends. The arrows connecting each action in the diagram is the control flow which show how the actions flow from one action to the other between the user and the system.

The activity diagram helps to simplify and improve any process by clarifying complicated use cases and illustrates a business or workflow between users and the system.

**Summary**

In many cases, most of the classic mistakes that lead to the failure of an IT project are related to people and processes. In order to minimize such occurrence, there is the need to break down the deliverables in the high-level project charter into the more detailed business requirements which is done in the analysis phase of the SDLC.Requirements modeling, data (domain) modeling, and process modeling constitute the three major activities in the analysis phase. This report focusses on the detailed description about the learning outcomes of the analysis phase of SDLC covering the domain model class diagram, concept map, and activity diagram presented in class. The first diagram is the domain model class diagram of the AI experiment “What we eat” and is used to show the things that the AI system will keep track of in order to function when playing the game. It contains the classes, attributes, object, associations, multiplicity, and multiplicity constraints. The second diagram is the concept map which depicts suggested relationships between eight concepts in SDLC. The third diagram is an activity diagram used to show the step required by the Wolfram Alpha’s system (and modules inside it) to deliver the given feature of currency conversion to a group of actors. The detailed descriptions of each diagram are discussed.

Prepared by: Eric Agyemang